MASTER EN SCIENCES INFORMATIQUES

Etude des protocoles de chiffrement et d'authentification 802,11

Vandaele, David

Award date:
2009

FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX, NAMUR
FACULTÉ D'INFORMATIQUE

ANNÉE ACADÉMIQUE 2008-2009

ÉTUDE DES PROTOCOLES DE CHIFFREMENT ET D'AUTHENTIFICATION 802.11

DAVID VANDAELE

Mémoire présenté en vue de l'obtention du grade de
Licencié en Informatique

Résumé

La norme 802.11 spécifie différents mécanismes de chiffrement et d'authentification ayant pour but d'offrir un service de sécurité mis en œuvre au niveau de la couche liaison d'un réseau Wi-Fi. Ce document traite de ces mécanismes, il en présente le fonctionnement détaillé, ainsi que les diverses attaques dont ils ont fait l'objet. Il propose ensuite une méthode de sélection du mécanisme de sécurisation 802.11 en fonction du contexte de travail.

Mots-clés: sécurité, Wi-Fi, WEP, WPA, 802.1X, 802.11, cryptanalyse.

Abstract

The 802.11 standard specifies various mechanisms of encryption and authentication with the aim of providing a security service implemented at the link layer of a Wi-Fi network. This document presents these mechanisms with accuracy and presents the various attacks on them. Then it proposes a method for selecting the best security mechanism for a 802.11 network depending on the context.

Keywords: Wi-Fi, WEP, WPA, 802.1X, 802.11, cryptanalysis.

Avant-propos

Je tiens à remercier :

Mr Jean-Noël Colin, promoteur de ce travail, pour son suivi, ses conseils et sa disponibilité.

Ma famille, particulièrement mon épouse, pour sa patience et son soutien tout au long de ces études.

Mon employeur pour la flexibilité dont il a fait preuve durant ces années d'étude.

Les personnes qui ont relu ce document.

Les membres du jury pour l'attention portée à ce travail.

TABLE DES MATIERES

TABLE DES FIGURES

Glossaire

AES : algorithme de chiffrement symétrique par bloc.

Algorithme de chiffrement asymétrique : algorithme de chiffrement basé sur une paire clé publique, clé privée.

Algorithme de chiffrement symétrique : algorithme de chiffrement basé sur un secret partagé.

AP (Access Point) : dispositif Wi-Fi servant de passerelle entre les stations et optionnellement le réseau filaire.

Authentification : processus visant à identifier de façon formelle, une personne, un périphérique ou une information.

BSS (Basic Service Set): ensemble de services de base d'un réseau Wi-Fi constitué d'une ou plusieurs stations connectées à un point d'accès.

CCMP (Counter Mode with CBC-Mac Protocol): algorithme de chiffrement et de contrôle d'intégrité utilisant AES.

CRC (Cyclic Redundancy Check) : algorithme de détection d'erreurs.

EAP (Extensible Authentication Protocol) : définit une structure de protocoles d'authentification extensible.

EAPOL (EAP Over Lan): extension de EAP définissant de nouveaux types de trames.

MIC (Message Integrity Code) : code d'intégrité destiné à protéger certains champs du message.

PMK (¨Pairwise Master Key): clé servant de base à la dérivation des clés de chiffrement TKIP et CCMP.

RC4 : algorithme de chiffrement symétrique par flux utilisé dans WEP.

SSID (Service Set Identity) : identifiant d'un ensemble de services.

Station (ST) : périphérique pouvant se connecter à un point d'accès au moyen d'un contrôleur Wi-Fi.

TK (Temporal Key) : clé temporaire dérivée de la clé PMK servant pour le chiffrement avec TKIP et CCMP.

TKIP (Temporal Key Integrity Protocol) : protocole de chiffrement utilisé dans WPA et basé sur RC4.

Wi-Fi (Wireless Fidelity) : définit les périphériques compatibles avec la norme 802.11.

WEP (Wired Equivalent Privacy) : premier protocole de chiffrement 802.11 basé sur l'algorithme RC4.

WPA (Wi-Fi Protected Access) : protocole de chiffrement 802.11 basé sur l'algorithme TKIP.

INTRODUCTION

Les réseaux Wi-Fi se sont répandus aussi bien dans les organisations que chez les particuliers durant ces dernières années. Or il s'avère que certains protocoles de sécurité exploités par ces réseaux ne sont pas efficaces.

De part ce travail, nous souhaitons présenter les enjeux de sécurité spécifiques à la mise en œuvre d'une communication Wi-Fi ainsi que les solutions de chiffrement et d'authentification 802.11 existantes à ce jour. Nous souhaitons ensuite réaliser un outil d'aide à la décision permettant d'opter pour l'une de ces méthodes de sécurisation en fonction du contexte de travail.

Afin de réaliser notre objectif, nous avons appliqué une méthodologie en 4 étapes :

1. Présentation des caractéristiques et des enjeux d'une communication Wi-Fi.

2. Présentation des notions de base de cryptologie nécessaires à la compréhension des solutions existantes.

3. Etat de l'art des solutions existantes.

4. Constitution et exploitation d'un outil d'analyse.

Notre recherche est basée sur des ouvrages scientifiques sélectionnés en fonction des différents domaines abordés. Ainsi, nous avons constitué une bibliographie principalement axée sur la cryptographie, la cryptanalyse et la norme 802.11.
Nous avons aussi recherché des publications d'articles sur internet, traitant principalement des attaques menées sur les protocoles de chiffrement.

Nous avons, dans le premier chapitre, présenté les enjeux spécifiques aux réseaux Wi-Fi ainsi que la norme 802.11.

Le chapitre 2 apporte les notions de cryptologie nécessaires pour aborder les chapitres 3 et 4. Ces derniers développent les mécanismes de chiffrement et d'authentification existants et décrivent les attaques menées sur ces protocoles.

Le chapitre 5 présente une méthode destinée à faciliter le choix d'une solution de sécurisation.

Chapitre 1 : Caractéristiques et enjeux d'une communication Wi-Fi

1.1 Architecture des réseaux Wi-Fi

1.1.1 Définition et justification

Les réseaux Wi-Fi répondent à un besoin de mobilité pour les utilisateurs souhaitant se connecter à un réseau d'entreprise ou domestique.

Ce besoin de mobilité résulte notamment de l'émergence de différents types de périphériques mobiles de plus en plus présents dans notre quotidien (Ordinateurs portables, PDA, ...). Outre la mobilité, un réseau Wi-Fi présente une solution à moindre coût pour la mise en place d'un réseau dans un bâtiment non câblé, ou pour relier deux réseaux existants. Nous verrons par la suite que si un réseau Wi-Fi présente des avantages en termes de coût et de souplesse d'utilisation, il amène certaines contraintes en termes de sécurité. De plus, il ne peut actuellement pas offrir des débits comparables aux réseaux filaires ou optiques. (Le Wi-Fi atteint des débits de l'ordre de 54 Mb/s, une connexion Ethernet filaire classique a un débit de 100 Mb/s tandis qu'une connexion en fibre optique peut atteindre un débit de 10 Gb/s)

1.1.2 Architecture

Les réseaux Wi-Fi sont constitués de différents acteurs dont l'agencement détermine l'architecture du réseau.

1. Les stations : représentent les terminaux entre lesquels les informations sont échangées au sein du réseau Wi-Fi.

2. Le point d'accès (PA) : un point d'accès sert d'intermédiaire entre un réseau filaire et le réseau Wi-Fi.

3. Le système de distribution (DS) : permet d'interconnecter différents points d'accès afin d'augmenter la portée du réseau Wi-Fi. Généralement ces points d'accès sont reliés via Ethernet.

Réseaux indépendants (IBSS : Independent Basic Service Set)

L'architecture la plus simple est constituée - comme illustré à la figure ci-dessous - de stations indépendantes communiquant entre elles. Cette architecture est aussi appelée réseau « ad hoc ».

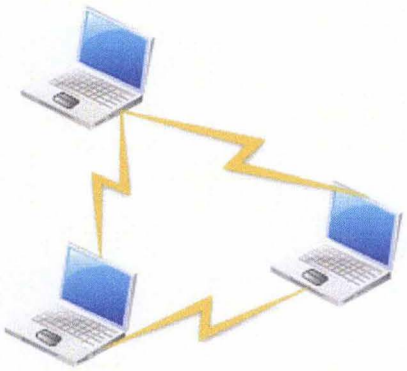

Figure 1: IBSS

L'aire de services de base est représentée par le rond plus foncé. Pour être associée au réseau, une station doit au minimum se trouver dans l'aire de services de base.

Réseaux infrastructure (*BSS Basic Service Set)*

L'architecture BSS contient un point d'accès et une ou plusieurs stations qui y sont connectées. Cet ensemble forme l'unité élémentaire d'un réseau Wi-Fi, appelé Basic Service Set. Un BSS est identifié par un identifiant unique appelé BSSID
Comme - illustré à la figure suivante - le point d'accès sert d'intermédiaire pour toutes les communications. Une station A désirant communiquer avec une station B le fait par l'intermédiaire du point d'accès AP.

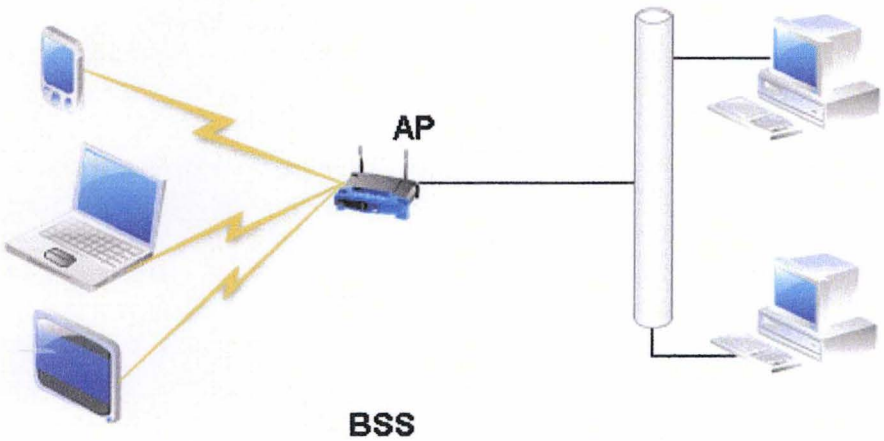

Figure 2:BSS

Ce point d'accès permet aussi une connexion vers le réseau filaire. L'aire de services de base est délimitée par la zone grisée.

Aires de services étendues (*Extended service set)*

Une telle architecture permet de couvrir des zones plus importantes qu'une architecture BSS. Le principe est de relier au moyen du système de distribution, plusieurs points d'accès auxquels est associé un même SSID. Les différentes aires de services de base vont - comme illustré à la figure suivante - se chevaucher et ainsi former une zone étendue.

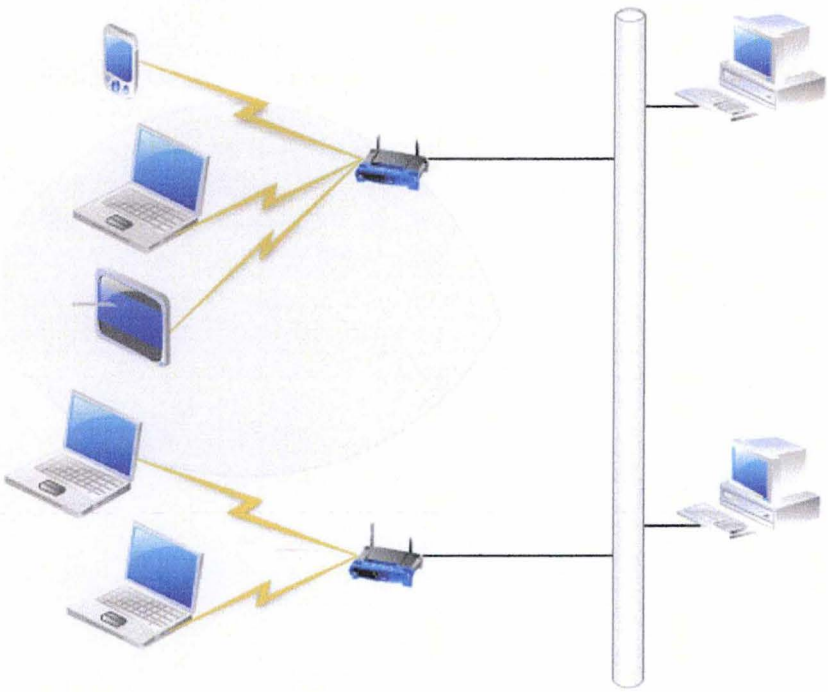

Figure 3: ESS

Les stations se trouvant dans cette zone étendue pourront alors communiquer entre elles. Les stations auront la possibilité de se déplacer dans l'ensemble de la zone, tandis que l'association au point d'accès le plus proche sera automatiquement gérée.

1.2 Norme 802.11 et Wi-Fi Alliance

La norme 802.11 définie par l'IEEE standardise la définition des réseaux Wi-Fi. Les normes IEEE 802 couvrent la couche physique et la couche liaison de données du modèle OSI. La figure ci-dessous illustre la place de la norme 802.11 dans la famille 802.

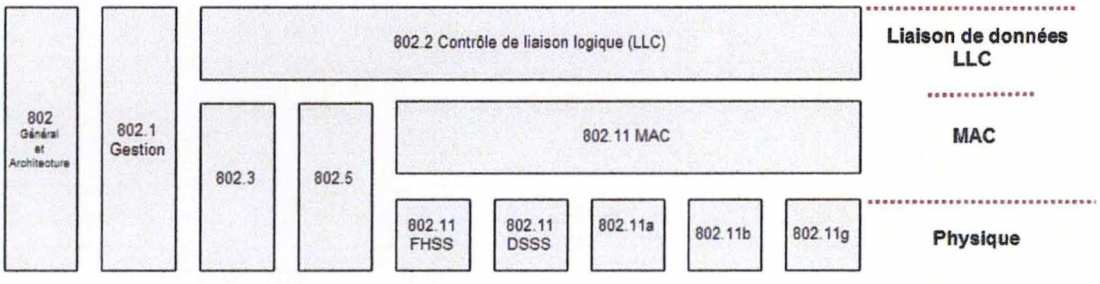

Figure 4: famille 802.11 [GAST, 2005]

La WECA, (Wireless Ethernet Compatibility Alliance) créée en 1999, a pour but de certifier le matériel répondant aux exigences de la norme 802.11. C'est ainsi que le label Wi-Fi (Wireless-Fidelity) est apparu. La WECA a ensuite été renommée en Wi-Fi Alliance.[Atelin, 2008]

A titre informatif, le 802.3 représente les réseaux Ethernet tandis que le 802.5 représente les réseaux Token Ring.

1.2.1 La couche physique

Comme illustré à la figure ci-dessous la norme 802.11 reprend différents standards concernant la couche physique (802.11a, 802.11b, 802.11g) qui se distinguent notamment par les bandes de fréquence utilisées, les débits proposés et les techniques d'étalement du spectre. La couche physique de la norme 802.11 dépasse la portée de ce travail.

1.2.2 La couche liaison de données

Pour rappel, cette couche a pour but de transférer un datagramme en provenance de la couche réseau d'un nœud à un autre nœud adjacent en l'encapsulant dans une trame MAC.

La couche liaison de données est subdivisée en deux sous-couches :

1. LLC (*Logical Link Control)* : définie par le standard 802.2 et commune à l'ensemble des réseaux locaux. Elle assure notamment le contrôle d'erreur, la gestion des retransmissions…

2. MAC (*Media Access)* : transmet les informations à la couche physique. C'est à ce niveau que l'accès au média est géré. Les différents processus d'authentification et de chiffrement que nous développerons dans ce travail prennent place à ce niveau. Il est possible d'assurer la sécurité d'une communication Wi-Fi au moyen des couches supérieures notamment avec IPSEC, ou avec la mise en place d'un VPN. Nous n'étudierons ici que les mécanismes spécifiques à 802.11.

1.2.3 Type et format des trames

De part la nature du média utilisé et le support de ce média (l'air), la couche MAC du 802.11 doit supporter un certain nombre de contraintes, comme les interférences, l'atténuation du signal, des problèmes de détection de l'occupation de la bande. En réponse à ces différentes contraintes, la couche MAC fait usage de trois types de trames définies dans les points suivants.

Trames de données (MSDU *Mac Service Data Unit)*

Ce type de trame encapsule les données des couches supérieures. Nous allons parcourir la structure générale de la trame sans entrer dans les détails. [Gast, 2005]
Elle est composée d'un en-tête MAC, des données en provenance des couches supérieures et d'un dernier champ destiné au contrôle d'erreurs.

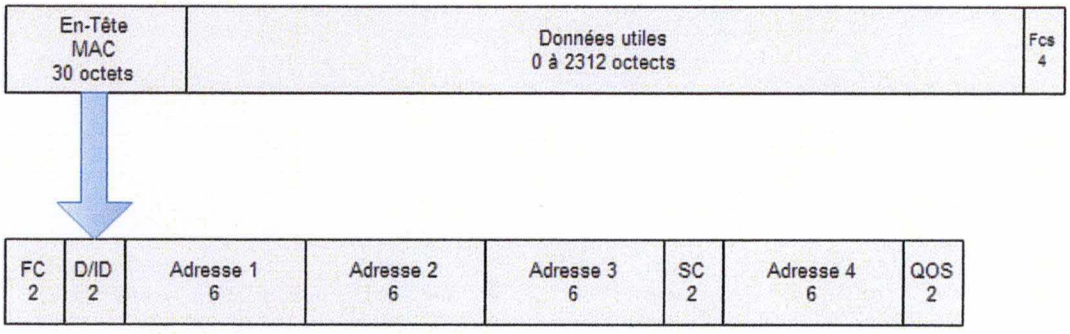

Figure 5: Structure de trame

L'en-tête MAC est - comme illustré ci-dessus - composé de :

- **FC (Frame control)** : contient diverses informations de contrôle, notamment sur le fait que le corps de trame est chiffré ou non et la direction de l'information, le type et le sous-type de la trame.

- **D/ID(Duration/ID)** : informe de la durée d'émission estimée de la trame, cette information intervient dans le mécanisme de réservation de la bande passante.

- Les trames contiennent quatre **champs d'adressage** utilisés en fonction du contexte et contenant les adresses source et destination des stations et les adresses source et destination des points d'accès (appelées respectivement adresse émetteur et récepteur).

- **SC (Sequence Counter)** : contient le numéro de séquence de la trame et le numéro de fragment.

- **QOS** : pour la gestion de la qualité.

Trames de gestion (MMPDU Mac Management Protocol Data Unit)

Ce type de trames sert à l'intégration des stations dans le réseau. Nous distinguons différentes trames de gestion :

1. Trames de balise : ces trames sont émises à intervalles réguliers (de l'ordre du dixième de seconde) par les points d'accès afin de signaler leur présence et contiennent certaines informations conditionnant l'accès au réseau.

2. Trames de requête et réponse d'association : dans le but d'associer une station au réseau.

3. Trames de désassociation et réassociation.

4. Trames de requête de sondage et de réponse de sondage : utilisées pour rechercher un réseau donné.

5. Trames d'authentification et de désauthentification.

Trames de contrôle

Ce type de trames assure le partage de la bande passante. Ce sont des trames de petite taille afin de limiter le risque de collision. Nous distinguons deux types de trames de contrôle :

1. RTS (Request To Send) : ces trames peuvent être envoyées par une station vers le point d'accès afin de savoir si elle peut disposer de la bande passante.

2. CTS (Clear To Send) : ces trames sont renvoyées à l'ensemble des stations pour spécifier que la station ayant émis la trame RTS peut utiliser la bande passante pour un intervalle de temps donné.

1.3 Principes de sécurité et risques spécifiques au Wi-Fi

1.3.1 Principes de sécurité

La sécurité d'un réseau est basée sur quatre axes ; une tentative d'attaque consistera à contourner l'un des ces axes [Atelin, 2008] :

1. L'**authentification** : assure que les acteurs d'un échange d'informations sont bien ceux qu'ils prétendent être.

2. La **confidentialité** : assure que les informations échangées ne seront pas interceptées par un élément externe auquel elles ne sont pas destinées.

3. L'**intégrité** : assure que les informations ne sont pas altérées durant l'échange.

4. La **disponibilité** : assure que le réseau soit utilisable.

La sécurité d'un réseau informatique peut être mise en place au niveau de toutes les couches du modèle OSI. Nous avons, par exemple, HTTPS au niveau de la couche application, IPSEC au niveau de la couche réseau, WEP et WPA au niveau de la couche liaison de données. Il est possible d'assurer la sécurité des couches inférieures en utilisant des mécanismes des couches supérieures. Ainsi la mise en place d'un réseau virtuel privé peut pallier à des failles de sécurité Wi-Fi. Ce travail ne s'intéresse cependant qu'aux techniques de sécurité du 802.11. Selon ce principe, la sécurité 802.11 est responsable de l'authentification, de la confidentialité, de l'intégrité et de la disponibilité de nœud à nœud. Elle n'assure en rien la sécurité de l'ensemble de la communication qui dépend des couches supérieures.

1.3.2 Sensibilité du Wi-Fi

Ce point a pour but de mettre en évidence les différents types de risques encourus par un réseau Wi-Fi sans entrer dans le détail des attaques.

Ecoute

Un réseau Wi-Fi, contrairement au réseau filaire, n'a pas de limitation physique bien déterminée. Dans un réseau filaire le support implique que l'écoute des communications nécessite un accès au bâtiment et une connexion physique sur le support. Dans un réseau Wi-Fi l'écoute peut avoir lieu en-dehors du bâtiment et sans la moindre connexion physique au réseau. Il faut donc considérer dans l'établissement d'une architecture Wi-Fi que l'ensemble des communications sera visible pour tout le monde, c'est pourquoi il faut mettre en place des mécanismes de chiffrement.

Nous pouvons distinguer deux types d'écoute :

1. **Ecoute passive** : consiste à analyser les trames transitant sur le réseau dans le but d'obtenir des informations. Dans un réseau Wi-Fi ce type d'attaque est quasiment indétectable.

2. **Ecoute active** : consiste à injecter des informations au sein d'un réseau afin de solliciter une réponse à analyser (par exemple pour retrouver une clé de chiffrement). Les deux types d'écoute mettent en danger le principe de confidentialité.

Une technique appelée « Wardriving » consiste à rechercher les différents réseaux Wi-Fi dans un certain périmètre. Le « Warchalking » consiste à marquer la place de chaque réseau ainsi identifié par un sigle spécifiant son niveau de sécurité. Un exemple de ces sigles est illustré dans la figure ci-dessous [Atelin, 2008].

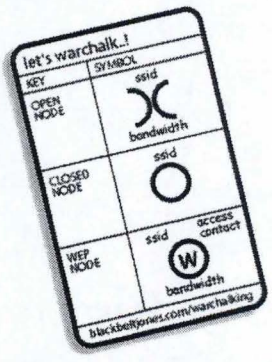

Figure 6: Warchalking [Atelin, 2008]

Déni de service

Le déni de service s'attaque à la disponibilité du réseau. Les ondes électromagnétiques sont sensibles aux interférences. Certains dispositifs permettent d'émettre des ondes dont le but est de produire des interférences rendant ainsi l'utilisation du réseau Wi-Fi impossible. Certaines techniques mettent en œuvre un point d'accès illégitime (axe d'authentification) qui émettra des requêtes forçant les stations à se déconnecter, rendant ainsi le réseau inutilisable.

Intrusion

L'intrusion consiste à obtenir un accès au réseau Wi-Fi de façon illégale. Différentes techniques basées sur l'analyse de trafic ou utilisant un point d'accès illégitime permettent d'obtenir un accès au réseau. Ce type d'intrusion peut aussi bien être utilisé pour obtenir un simple accès à internet que pour mener des actions illégales de façon anonyme.

1.4 Les différentes étapes d'une communication Wi-Fi

Nous allons spécifier une vue générale des différentes étapes d'une communication Wi-Fi en soulignant les différentes alternatives possibles.
Nous partirons de l'exemple illustré à la figure suivante, dans lequel Alice (Ordinateur A) souhaite envoyer un message M à Bob (Ordinateur B).

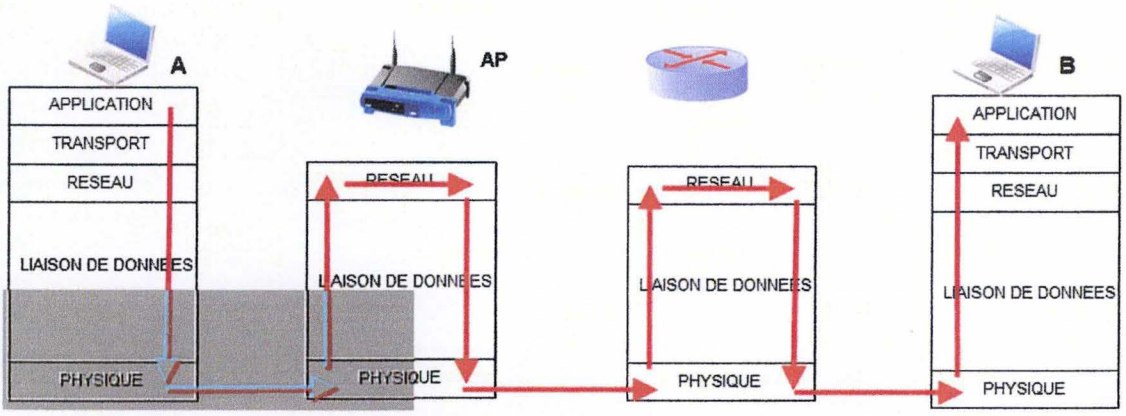

Figure 7: Communication Wi-Fi

Dans ce scénario Alice se connecte au moyen d'un adaptateur sans fil à un point d'Accès (AP). Tandis que Bob se connecte au moyen d'un réseau filaire.
Le dialogue 802.11 ne concernera que la partie grisée de la figure ci-dessus, ce qui comprend :

- La partie MAC de la couche liaison de données entre A et AP. Le dialogue entre AP et PB concerne la norme 802.2.
- La couche physique entre A et AP.

La suite de la transmission du message vers Bob, utilisera d'autres protocoles de différentes couches comme Ethernet,TCP, IP, HTTP…

Dans le cadre de ce travail, nous nous intéresserons à la couche liaison de données de la norme 802.11.

Afin de transmettre son message, Alice devra préalablement associer sa station de travail auprès de son point d'accès. Une telle association est nécessaire pour tout nœud 802.11 du réseau et requiert différentes étapes illustrées par le diagramme d'activités ci-dessous.

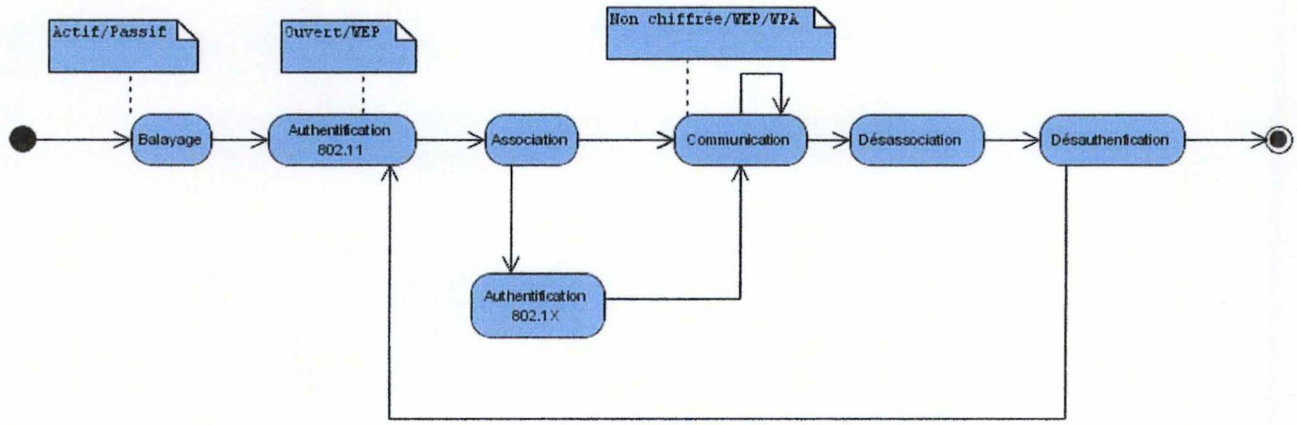

Figure 8: Etapes du dialogue 802.11

1.4.1 Balayage

Cette activité consiste à rechercher les différents points d'accès disponibles dans le périmètre de la station de travail. Le balayage peut être paramétré afin de n'afficher que les réseaux répondant à certains critères, par exemple: le SSID, la liste des canaux à scanner..
On distingue deux types de balayage :

1. **Actif** : ce mode est notamment utilisé lorsqu'un point d'accès n'émet pas de trames balises. La station de travail cherche à se connecter à un réseau particulier et envoie une trame de type « Probe Request » contenant le SSID. Si le point d'accès concerné reçoit une telle trame, il manifeste sa présence avec une trame de type « Probe Response ».

2. **Passif** : la station de travail se met en écoute sur les différents canaux afin d'intercepter les trames balises émises par le point d'accès.

Suite à l'activité de balayage, un rapport contenant la liste des réseaux disponibles avec leurs caractéristiques est générée. Alice doit alors sélectionner le réseau auquel elle souhaite se connecter. Il est aussi possible de paramétrer la station de travail afin que le choix du réseau se fasse automatiquement.

1.4.2 Authentification 802.11

Afin de s'associer au réseau sélectionné à l'étape précédente, la station de travail d'Alice doit être authentifiée auprès du point d'accès. A ce stade, nous parlons d'une authentification de bas niveau, qui est imposée par la norme 802.11 [GAST, 2005] et qui se présente sous deux formes (cf. Chapitre 4) :

1. **Ouverte** : qui correspond à l'authentification nulle. Alice se présente et le point d'accès l'accepte sans contrôle.

2. **Partagée** : ce mode d'authentification impose le partage d'un secret entre le point d'accès et la station de travail. Le point d'accès vérifie via un mécanisme de « Challenge-Response » si la station de travail est bien en possession du secret, si oui il l'authentifie, sinon il la rejette.

1.4.3 Association

Une station de travail authentifiée auprès d'un point d'accès peut émettre une demande d'association vers celui-ci. Si la demande est acceptée, elle se traduira par la prise en charge de toutes les trames à destination ou provenant de la station de travail par le point d'accès. La station de travail recevra un identifiant d'association « AID ».
Le diagramme de séquence suivant [GAST, 2005] illustre le processus d'association.

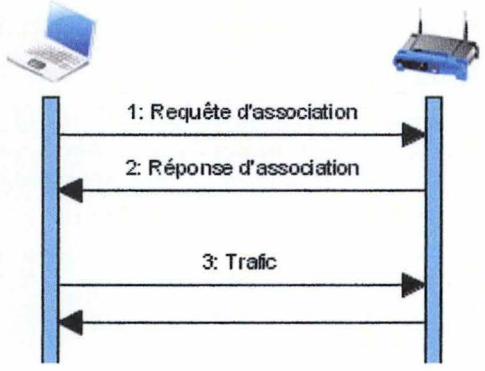

Figure 9: Association

Une station ne peut être associée qu'à un seul point d'accès simultanément.
Après l'association, l'échange de trafic « utile » entre le point d'accès et la station de travail peut démarrer. Ce trafic peut toutefois servir à mettre en place un mécanisme d'authentification plus robuste que celui imposé par la norme 802.11, comme le 802.1 X. (cf. 4.4 Authentification 802.1X)

1.4.4 Communication

L'activité de communication consiste à mettre en œuvre l'échange de « données utiles » entre la station de travail et le point d'accès. Lors de cette activité, le message d'Alice en provenance des couches supérieures sera transmis au nœud suivant.
C'est notamment à ce niveau que nous pouvons implémenter **les méthodes de chiffrement** proposées par la norme 802.11 (WEP, WPA, WPA2 …)

1.4.5 Désassociation

Une station peut émettre une demande de désassociation lorsqu'elle désire quitter le réseau ou lorsque qu'elle désire s'associer à un autre point d'accès (réassociation).

1.4.6 Désauthentification

De façon analogue à la désassociation cette activité permet de clôturer une relation d'authentification.

Chapitre 2 : Notions de base de cryptologie

2.1 Cryptologie

« Cryptology is the art and science of making and breaking « secret codes »
[M.LOW, 2007]

La cryptologie peut se définir comme la science qui consiste à transférer des informations de façon sécurisée [SWENSON, 2008].

La cryptologie - comme illustré dans la figure ci-dessous - est constituée de deux matières principales :

- **La cryptographie** qui consiste à générer des messages chiffrés appelés cryptogrammes.
- **La cryptanalyse**: qui consiste à « casser » ces cryptogrammes.

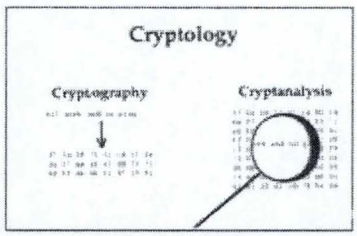

Figure 10: Cryptologie [SWENSON, 2008]

2.2 Cryptographie

Nous trouvons la définition de cryptographie suivantes :

«L'art et la science de garder le secret des messages »[Schneier, 2001].

La mise en œuvre de moyens cryptographiques permettra d'établir un échange d'informations (appelées message M) entre un expéditeur (communément Alice) et un destinataire (communément Bob) tout en garantissant :

1. L'intégrité : M n'a pas été altéré entre l'expéditeur et le destinataire.
2. La confidentialité : seul le destinataire pourra avoir accès à M.
3. La non-répudiation : garantit au destinataire que M provient bien de l'expéditeur.

La cryptographie utilise le procédé de chiffrement E afin de transformer M en un message incompréhensible appelé cryptogramme C.
Lors du processus de déchiffrement D, le destinataire détermine M à partir de C.
Les processus de chiffrement et déchiffrement sont mis en œuvre dans des algorithmes cryptographiques. Généralement ces algorithmes sont rendus publics et utilisent un paramètre tenu secret, appelé la clé K [1]. L'ensemble des valeurs pouvant être pris par la clé K est appelé « espace des clés ».

2.2.1 Algorithme à clé privée

Aussi appelé algorithme symétrique, ce type d'algorithme utilise une clé identique pour le processus de chiffrement et déchiffrement. Les algorithmes symétriques nécessitent que la clé soit connue de l'expéditeur, du destinataire et uniquement de ceux-ci, ce qui implique un échange préalable de clés.

$$E_k(M) = C$$
$$D_k(C) = M$$

Figure 11: Clé privée [Schneier, 2001]

Chiffrement de Vernam (One Time Pad)

Le chiffrement de Vernam existe depuis 1917[Schneier 2001] et garantit que les cryptogrammes émis sont inviolables sous réserve de respecter trois propriétés qui, selon Shannon, permettent d'obtenir le « secret parfait ».

1. La longueur de la clé doit être identique à longueur du message en clair.
2. La clé doit être générée de façon aléatoire.
3. La clé ne peut servir qu'une seule fois.

Le mécanisme de chiffrement mis en œuvre consiste à combiner le texte en clair et la clé par application du « ou exclusif ». Pour déchiffrer le message il faut refaire l'opération dans le sens inverse.

$$C = M \oplus K \text{ et } M = C \oplus K$$

Par la condition numéro 2, la clé est générée de façon aléatoire, ce qui rend toutes les clés équiprobables.

[1] Il existe des algorithmes restreints n'utilisant pas de clé et dont le secret repose sur le secret de l'algorithme lui-même. Ce type d'algorithme est cependant très peu utilisé.

Si toutes les clés sont équiprobables, alors pour un cryptogramme de longueur n, tous les messages de longueur n ont la même probabilité d'être le message d'origine. Ceci implique que, quelle que soit la puissance de calcul disponible, il est impossible de retrouver le message d'origine sur base du cryptogramme sans être en possession de la clé.

Bien que ce mécanisme de chiffrement soit infaillible, il n'en est pas moins difficile à appliquer. En effet, il faut pouvoir générer une clé aléatoire. Générer un nombre qui soit tout à fait aléatoire nécessite l'utilisation d'algorithmes qui reçoivent en entrée des paramètres du monde réel considérés comme aléatoires [Schneier, 2001] (bruit atmosphérique, émission radioactive...). Les algorithmes couramment utilisés généreront des séquences pseudo-aléatoires.

De plus, ces conditions représentent une difficulté au niveau de la distribution des clés. En effet, la clé doit être de même longueur que le message à chiffrer et connue à la fois de l'expéditeur et du destinataire du message, ce qui nécessite de mettre en œuvre un mécanisme d'échange préalable d'une clé aussi longue que le message à transmettre... (Comment garantir que cet échange n'ait pas été intercepté et que la clé soit toujours intègre ?).

Ensuite, il faut pouvoir reconstruire et échanger une nouvelle clé pour chaque message. En effet, posséder deux messages chiffrés à partir d'une même clé suffit pour obtenir le message en clair via des techniques d'analyse de ressemblance. Ce fut le cas par exemple dans le cadre du projet « VENOA » qui a permis à la NSA de décrypter certaines communications d'agents soviétiques durant la Guerre froide [Martin, 2004].

Etant donné les difficultés de mise en œuvre et bien qu'infaillible, cette technique ne sera utilisée que dans des contextes où l'on souhaite assurer une confidentialité totale en dépit de la rapidité d'échange. Pour l'histoire, ce mécanisme était utilisé dans « le téléphone rouge » entre les Etats-Unis et l'ex-Union soviétique.[Schneier, 2001]

Chiffrement en continu

Un algorithme utilisant le mode de chiffrement en continu travaille bit par bit. Le texte en clair est combiné avec des bits aléatoires par un « ou exclusif » pour générer le cryptogramme. Toute la complexité réside dans le générateur de bits aléatoires qui, comme mentionné au point précédent, généreront des séquences pseudo-aléatoires. Le générateur de bits pseudo-aléatoires est caractérisé par un état interne et une fonction de sortie. Son mode opératoire est le suivant :

1. Initialisation de l'état interne en fonction d'une clé d'initialisation.

2. Pour chaque bit de texte en clair répéter :
   a. Exécuter la fonction de sortie : sélectionner un bit à partir de l'état interne.
   b. Calculer un nouvel état interne.

Nous distinguons deux types de chiffrements en continu :

1. Chiffrement autosynchrone en continu : l'état interne dépend d'un nombre de bits du texte chiffré qui précède. L'intérêt est de pouvoir synchroniser le générateur de bits aléatoires de façon automatique lors du déchiffrement après avoir reçu n bits chiffrés [Schneier, 2001]. Ce type de chiffrement est résistant aux pertes de bits mais une altération d'un bit se répercutera sur n bits.

2. Chiffrement synchrone en continu: le flux de bits aléatoires est calculé de façon indépendante du flux du message. La perte d'un bit va générer une désynchronisation totale du message tandis que l'altération d'un bit se répercutera uniquement sur un bit.

Chiffrement par bloc

Le chiffrement par bloc concerne une classe d'algorithmes travaillant sur des blocs de bits en clair de longueur fixée et génèrant des blocs de textes chiffrés de même longueur. Un même bloc sera toujours chiffré de la même façon avec une même clé [Schneier, 2001]. (Nous verrons à ce sujet que l'on associe communément un compteur, appelé vecteur d'initialisation, à la clé afin d'éviter ce phénomène). Un algorithme de chiffrement par bloc peut être utilisé selon différents modes.

ECB (Electronic Code Book)

Nous pouvons comparer le mode ECB à un livre de codes faisant correspondre à un bloc de texte clair un bloc de texte chiffré. Un tel livre de codes est présent pour chaque valeur possible de la clé.[Swenson, 2008]
Ce mode est sensible aux attaques par bloc rejoué et aux analyses statistiques. Des séquences identiques de textes produiront des séquences identiques de textes chiffrés. Le mode ECB a cependant l'avantage de ne pas devoir être appliqué en suivant l'ordre du flux à chiffrer, ce qui permet notamment de distribuer le travail sur plusieurs processeurs.

Le chiffrement et le déchiffrement sont respectivement effectués de la façon suivante :

$$C_i = E_k(M_i)$$
$$M_i = D_k(C_i)$$

CBC (Cipher block chaining)

Avec le mode CBC, le chiffrement du bloc courant dépend des blocs précédemment chiffrés. On combine le cryptogramme obtenu pour le bloc précédent par un « ou exclusif » avec le texte en clair du bloc courant avant de le chiffrer. Afin d'éviter qu'un même message en clair ne génère un même cryptogramme, on ajoute un vecteur d'initialisation en début de message.

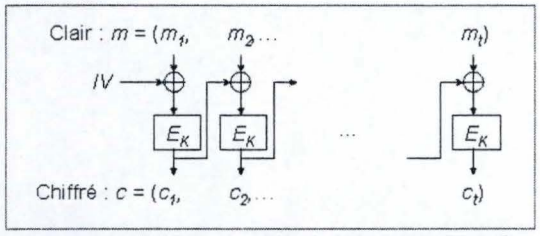

Figure 12: Mode CBC [PICSI, 2006]

$$C_i = E_k (M_i \oplus C_{i-1})$$
$$M_i = C_{i-1} \oplus D_k (C_i)$$

CFB (Cipher Feed Back)

Le mode de chiffrement CFB permet de commencer le processus de chiffrement avec des unités de taille « n », « n » étant plus petit qu'un bloc complet. Nous combinons un chiffrement autosynchrone en continu au mode CBC. Un registre à décalage de la taille d'un bloc est initialisé sur base d'un vecteur d'initialisation avant d'être entièrement chiffré. Les n bits de poids fort ainsi obtenus sont combinés avec les n bits suivants de texte clair au moyen d'un « ou exclusif ». Le résultat peut directement être transmis. Le résultat est ensuite inséré dans le registre à décalage avant de le décaler de n bits vers la gauche.

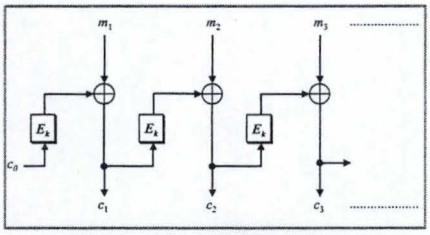

Figure 13: CFB [RSA]

$$C_i = M_i \oplus E_k(C_{i-1})$$
$$M_i = C_i \oplus E_k(C_{i-1})$$

OFB (Output Feed Back)

Le mode OFB est similaire au mode CFB excepté le fait que les bits qui sont réinjectés dans le registre à décalage le sont avant d'avoir subi l'opération XOR avec le texte en clair. L'avantage est de pouvoir effectuer une partie du travail avant même d'être en possession du texte en clair.

2.2.2 Algorithme à clé publique

Aussi appelé algorithme asymétrique, les algorithmes à clé publique utilisent une clé pour le chiffrement et une autre pour le déchiffrement. La clé de chiffrement est appelée clé publique (Kp), celle utilisée pour le déchiffrement est appelée clé privée (Ks). Ces deux clés sont liées entre elles de façon à ce que tout cryptogramme chiffré à partir de Ks ne soit déchiffrable qu'à l'aide de Kp et inversement. De plus Ks et Kp sont construites de façon à ce l'on ne puisse pas déterminer Ks à partir de Kp.
Tout cryptogramme obtenu à partir de la clé publique peut être déchiffré au moyen de la clé privée correspondante et réciproquement.
L'avantage de ce type d'algorithme réside dans la facilité de distribution des clés. Le chiffrement s'effectue à partir de la clé publique du destinataire qui est connue de tous (pas d'échange préalable de clés). Le déchiffrement ne peut s'effectuer qu'au moyen de la clé privée du destinataire, tenue secrète.

$$E_{Kp}(M) = C$$
$$D_{Ks}(C) = M$$

Figure 14 : Clé publique

Les algorithmes à clé publique sont notamment utilisés dans le mécanisme de signature électronique. Si Alice désire signer un message avant de l'envoyer à Bob, elle va générer une signature à partir d'une fonction de hachage « h » appliquée au message (cf. 2.2.3).
Elle va ensuite chiffrer ce résultat au moyen de sa clé privée pour produire la signature.
A la réception du message, Bob commence par déchiffrer la signature du message au moyen de la clé publique d'Alice. Il déchiffre ensuite le message et lui applique la fonction « h ». Si le résultat obtenu est le même que celui qu'il a déchiffré avec la clé publique d'Alice, il a la certitude que le message provient bien d'Alice et qu'il n'a pas été modifié.

$$E_{Ks}(h(M)) + E_{Kp}(M)$$

Figure 15: Signature

Les algorithmes à clé publique ont cependant le désavantage d'être plus lents que les algorithmes à clé privée. [Schneier, 2001]
Les algorithmes à clé publique seront souvent utilisés afin de distribuer les clés dans un algorithme à clé privée. Un tel système est appelé « Système hybride ».

2.2.3 Les fonctions de hachage

Les fonctions de hachage sont fréquemment utilisées dans des étapes intermédiaires des algorithmes cryptographiques, notamment dans la mise en œuvre du contrôle d'intégrité. Dans l'extrait ci-dessous [M. LOW 2007, page 193], nous rappelons certains principes caractérisant les fonctions de hachage :

- *« Compression : la sortie de la fonction de hachage doit être plus petite que son entrée, généralement cette sortie sera de taille fixe.*

- *Efficacité : la complexité de calcul ne doit pas croître de façon exponentielle en fonction de la taille de l'entrée.*

- *Sens unique : il doit être impossible de pouvoir calculer l'entrée sur la base de la sortie de la fonction.*

- *Sur la base d'une entrée et de sa sortie associée, il est impossible de calculer une autre entrée donnant la même sortie. Bien que de telles entrées puissent exister, il ne doit pas être possible de les calculer.*

- *Il est impossible de calculer une paire d'entrées différentes ayant une sortie identique. »*

Par opposition à une fonction linéaire, le moindre changement dans l'entrée de la fonction doit avoir une répercussion importante et non proportionnelle sur le résultat de la fonction [M. LOW, 2007].

Un exemple de fonction de hachage se trouve en Annexe 1.

2.3 Cryptanalyse

« *La cryptanalyse est la science de la reconstitution du texte en clair sans connaître la clef* »[Schneier, 2001].

«*Cryptanalysis is the study of defeating and strengthening cryptographic technics* »[Swenson, 2008].

Le but de la cryptanalyse est de retrouver un message en clair et/ou la clé à partir du message chiffré. La cryptanalyse est un outil permettant de mettre en évidence des failles dans les algorithmes cryptographiques ou dans la façon dont ils sont implémentés dans un contexte donné afin de les corriger.

Une attaque est une tentative de cryptanalyse [Schneier, 2001].

Lors d'une tentative de cryptanalyse, l'attaquant commet des attaques dans un contexte basé sur les axiomes suivants :

1. Le texte chiffré est toujours disponible pour l'attaquant, bien que dans la pratique il est possible que le texte chiffré ne soit pas aisément accessible.

2. Axiome de Kerckhoffs : l'attaquant connaît tous les détails de l'algorithme utilisés. [Schneier, 2001]. A nouveau, ce n'est pas toujours le cas dans la pratique mais il faut éviter de baser la sécurité sur l'aspect secret de l'algorithme qui, tôt ou tard, finira par être découvert. De plus, rendre les algorithmes publics permet, comme nous l'avons vu dans la définition de la cryptanalyse, de les confronter à différentes tentatives d'attaque et d'ainsi mettre en évidence les failles de sécurité à corriger.

Il existe deux types d'attaquants [Buchmann, 2006] :

1. **Passif** : cet attaquant réalise une écoute passive afin d'intercepter des messages chiffrés. Il se base uniquement sur les informations ainsi collectées pour tenter de décrypter le message.

2. **Actif** : cet attaquant réalise une écoute active, il a donc la possibilité d'émettre des trames afin de solliciter certaines réponses.

2.3.1 Attaque sur cryptogramme chiffré seul (Ciphertext-only)

Dans une attaque sur cryptogramme chiffré, le cryptanalyste tente de retrouver la clé ou le texte en clair en n'ayant aucune connaissance du message d'origine. Il ne dispose que du cryptogramme de plusieurs messages chiffrés avec le même algorithme mais avec des clés pouvant être différentes.
La mise en œuvre d'une telle attaque peut se faire par la recherche exhaustive/force brute. Dans ce cas la difficulté dépend principalement de la taille de l'espace des clés et de la puissance de calcul disponible. Une autre possibilité d'attaque sur texte chiffré seul est l'étude sur base des propriétés statistiques du langage du texte clair [Buchmann,2006].

2.3.2 Attaque à clair connu (known plaintext)

Dans une attaque à clair connu, l'attaquant dispose d'un nombre restreint de paires message/cryptogramme. Si le principe de chiffrement aléatoire, imposant qu'un même message chiffré à deux reprises produise deux cryptogrammes différents, n'est pas correctement appliqué à chaque occurrence du cryptogramme en possession de l'attaquant, celui-ci connaîtra automatiquement le message clair correspondant.

2.3.3 Attaque à clair choisi (chosen plaintext)

Dans une attaque à clair choisi, l'attaquant a la possibilité de choisir le texte en clair et d'obtenir le cryptogramme associé. Ceci permet à l'attaquant de choisir des textes en clair spécifiques qui livreront plus d'informations sur la clé [Schneier, 2001]. Les attaques à clair choisi sont caractéristiques des algorithmes à clé publique pour lesquels l'attaquant a accès à la clé de chiffrement. Comme pour les attaques à clair connu, on insiste sur l'importance du principe de chiffrement aléatoire.

2.3.4 Attaque à cryptogramme choisi

Dans une attaque à cryptogramme choisi, l'attaquant a la possibilité de décrypter le cryptogramme de son choix sans connaître la clé qu'il continue à rechercher [Buchmann, 2006].

2.3.5 Attaque par rejeu

Le principe d'une attaque par rejeu consiste à intercepter une trame particulière dans le trafic réseau (par exemple la trame autorisant à créditer un compte bancaire) et de la réinsérer dans le trafic afin qu'elle soit exécutée plusieurs fois.

2.3.6 Attaque par force brute

Une attaque par force brute consiste à essayer toutes les valeurs de l'espace de clés possibles afin de décrypter un message. Statistiquement, une telle attaque aboutira après avoir essayé la moitié des valeurs possibles.

CHAPITRE 3 : SOLUTIONS DE CHIFFREMENT EXISTANTES

Ce chapitre présente les solutions de chiffrement 802.11 existantes. Chaque solution est présentée sur base de la méthode suivante:

1. Présentation de haut niveau des différents composants.
2. Définition détaillée de chaque composant.
3. Présentation des interactions entre les différents composants.
4. Présentation des failles du système.

3.1 WEP

3.1.1 Présentation

Le protocole WEP, défini dans la norme 802.11, peut être mis en œuvre afin d'assurer le chiffrement des données et ainsi proposer une première solution de maintien de la confidentialité des données. La Figure 7: Communication Wi-Fi"- modifiée, ci-dessous - illustre à quel niveau se déroulent les opérations de chiffrement/déchiffrement.

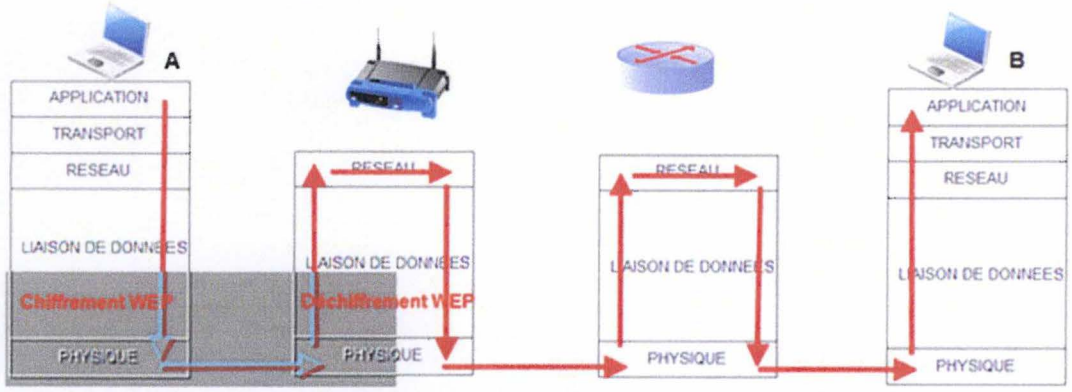

Figure 16: Communication WEP

Pour rappel le chiffrement se déroule après les phases d'authentification et d'association, durant l'activité de communication. (cf. Figure 8: Etapes du dialogue 802.11).

La figure ci-dessous donne un aperçu général du mécanisme de chiffrement WEP(sur base de [Gast, 2005]).

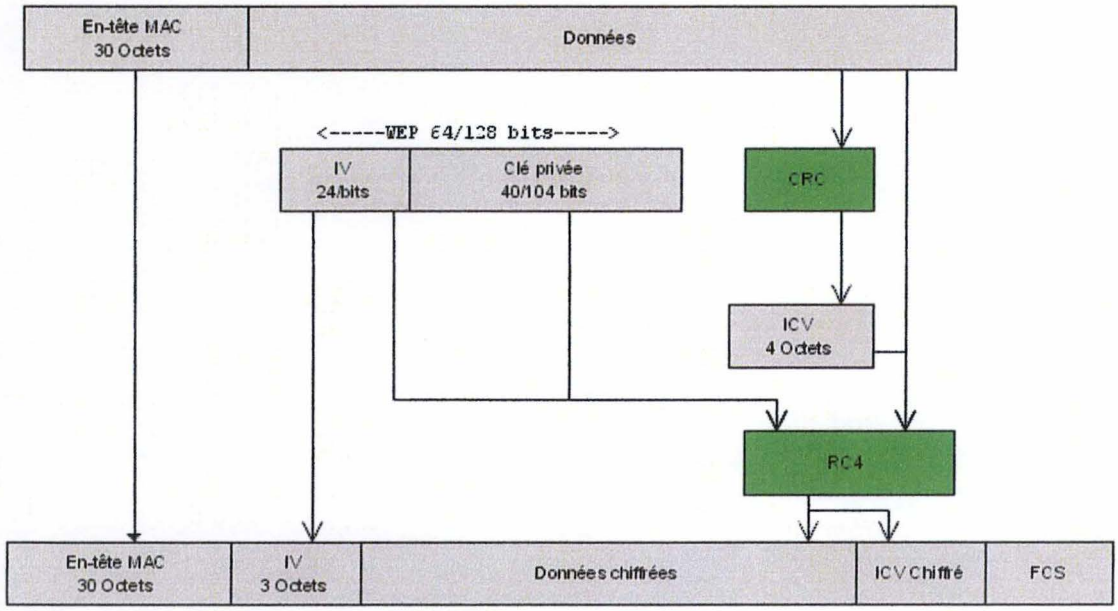

Figure 17: WEP

Comme illustré, la partie donnée de la trame MAC, correspondant aux données de la couche réseau, est chiffrée au moyen de l'algorithme RC4, tandis que l'intégrité des données est assurée au moyen du contrôle CRC.

Algorithme RC4

L'algorithme RC4 a été développé par Ron RIVEST en 1987 pour la société RSA. Il a été rendu public après avoir été diffusé anonymement sur une liste de diffusion « Cypherpunk » en 84 [Schneier, 2001].
C'est un algorithme à flux continu dont le principe est de générer une clé aléatoire de même longueur que le message.

RC4 est constitué d'une phase d'initialisation durant laquelle il utilise une clé secrète afin de préparer une table d'état « S », cette phase est appelée « KSA »(Key Scheduling Algorithm), et d'une phase de génération durant laquelle un octet est choisi de façon aléatoire dans la table S afin de chiffrer l'octet courant dans le flux du message M, phase appelée « PRGA » (Pseudo Random Generation Algorithm)[Kesley, 1996].

Initialisation

RC4 peut être utilisé en mode 8 ou 16 bits que nous généraliserons en n bits.
Dans la phase d'initialisation nous allons construire la table d'état S.
Cette table sera constituée de 2^n entrées de n bits et contiendra en permanence les valeurs allant de 0 à 2^n-1.

Dans un premier temps ces valeurs seront ordonnées, tandis qu'après la phase d'initialisation la table S représentera l'une des 2^n! permutations possibles. La permutation ainsi obtenue dépend d'une clé secrète K de longueur k, de deux compteurs i et j et d'un calcul modulo illustré dans la figure ci-dessous.

```
j :=0 ;
for(i=0 -> n-1)
        Si=i ;
for(i=0 -> n-1){
        j=(j + Si + K[i modulo k]) modulo n ;
        Echanger(Si,Sj);}
```

Figure 18: KSA

Génération du flux aléatoire et chiffrement

Dans la phase de génération du code, une entrée de n bits dans la table S sera sélectionnée pour être combinée par un « ou exclusif » aux n bits correspondants dans le flux du message M pour former n bits du cryptogramme C. La table S est alors permutée avant de sélectionner les n bits suivants.

```
i,j:=0;
m:=Mi;
while(m!=null){
        i:=(i+1)modulo n;
        j:=(j + Si) modulo n;
        Echanger(Si,Sj);
        Ci= S[(Si+Sj)modulo n]⊕ m ;
        m:=Mi;}
```

Figure 19: PRGA

Nous constatons que RC4 est une application du chiffrement de Vernam. Cependant le chiffrement RC4 n'est pas infaillible tel que nous le verrons dans la suite de ce travail. Ceci implique qu'au moins l'une des conditions de Shannon n'a pas été respectée. La taille de la clé étant égale à la taille du message M, les attaques sur RC4 exploiteront le fait que d'une part la clé générée est pseudo-aléatoire et non totalement aléatoire et d'une autre part, qu'il est possible d'utiliser deux fois la même clé.

Contrôle d'intégrité CRC

L'ICV (Integrity, Check, Value) de 32 bits ajouté en fin de trame correspond au contrôle d'intégrité CRC-32 de la trame WEP non chiffrée.
Le contrôle d'intégrité CRC-32 se base sur le polynôme générateur G suivant [Kurose, 2003]:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

Si r est le degré du polynôme générateur (32),
si M est le message en clair,
alors R est le reste de la division de $(M * 2^r)/G$ et correspond au contrôle d'intégrité ajouté à la suite de M.

Etant donné que chaque bit composant M intervient dans le calcul de R, une erreur dans un bit de données sera automatiquement détectée [Bryant, 2009].

Dans le cadre du protocole WEP, le reste R est ajouté au message M.
L'algorithme RC4 est ensuite appliqué à M+R pour calculer C, sur base d'une clé constituée du vecteur initial IV et de la clé secrète partagée K.
Finalement la trame constituée de IV + C est envoyée au destinataire.

Lors de la réception du message le reste de la division de (M+R)/G doit être égale à 0. Dans le cas contraire une erreur est détectée.

Un exemple est présenté en Annexe 3.

Interaction entre les composants

Le protocole WEP fait usage de l'algorithme RC4 afin de chiffrer les données transitant entre un point d'accès et un ordinateur distant.
Le protocole nécessite une clé secrète statique K de 40 ou 104 bits qui doit être enregistrée au niveau du point d'accès ainsi que sur chaque ordinateur client souhaitant s'y connecter.

Cette clé sera utilisée dans la phase d'initialisation de l'algorithme RC4. Cependant, afin d'éviter d'utiliser plusieurs fois la même clé, à chaque nouvelle trame, un vecteur d'initialisation IV de 24 bits sera ajouté à la clé K. Ceci permettra aussi d'éviter que deux messages ne génèrent un même cryptogramme. La clé devant être connue des deux parties, le vecteur d'initialisation sera transmis en clair dans la trame WEP. [Atelin, 2008]

Une trame WEP sera constituée - comme illustré ci-dessous - :

- Du vecteur d'initialisation en clair.
- D'un numéro de clé : les points d'accès peuvent généralement mémoriser quatre clés, ce numéro spécifie la clé utilisée.
- Des données chiffrées.
- Du contrôle d'intégrité chiffré (Integrity Check Value).

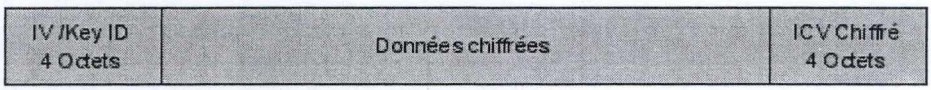

Figure 20: Trame WEP [Atelin, 2008]

3.1.2 Failles

Bien que le protocole WEP puisse sembler fiable, il existe de nombreuses failles notamment dues à la façon dont l'algorithme RC4 est utilisé ainsi qu'à l'utilisation du contrôle d'intégrité CRC-32 qui n'est pas suffisant pour détecter une modification volontaire.

La principale faiblesse du protocole WEP provient du vecteur d'initialisation sur 24 bits. En effet, cette valeur ne permet pas de créer un nombre suffisant de clés. Le vecteur d'initialisation étant transmis en clair, l'observation du trafic réseau permettra d'identifier deux paquets ayant des vecteurs égaux. [Atelin, 2008]

Ce point décrit les principales attaques menées sur le protocole WEP. Cette liste n'est pas exhaustive mais tente de faire apparaître les différents principes utilisés pour mener une attaque. Certaines variantes des attaques décrites ci-dessous permettent de les optimiser.

Force brute

L'attaque par force brute consistant à essayer tout l'espace de clés est réalisable pour des clés WEP de 40 bits. De plus, certains logiciels de configuration réduisent l'espace de clé en proposant de construire la clé WEP sur base d'une phrase de passe. L'algorithme utilisé afin de transformer cette phrase en clé peut, dans certains cas, réduire l'espace de clés réellement utilisé. [Fluhrer, 2001]

Attaque FMS

Du nom de leur auteur Fluhrer, Martin et Shamir, cette attaque est basée sur certaines propriétés de RC4. [Fluhrer, 2001] « *En effet pour certaines classes de clés une petite partie de la clé secrète détermine un grand nombre de bits de la permutation initiale* » [Fluhrer, 2001]. Ceci ayant pour conséquence que les premiers bits du flux chiffrant ne sont influencés que par un petit nombre de bits de la clé. Cette attaque a été l'attaque de référence sur base de laquelle la plupart des autres se sont basées, c'est pourquoi nous la détaillons ci-dessous (au moyen d'un exemple issu de [M. LOW, 2007]).

Soit un vecteur d'initialisation VI d'une taille de 3 octets de la forme suivante (ou V représente un octet connu):

Tableau 1: VI

i	0	1	2
Si	3	255	V

La table d'état S :

Tableau 2: S

i	0	1	2	3	4	5	…	255
Si	0	1	2	3	4	5	…	255

La clé K composée du vecteur d'initialisation VI et de la clé secrète sur x bits :

Tableau 3: K

i	0	1	2	3	4	5	…	255
Si	3	255	V	K	K	K	…	K

L'étape d'initialisation KSA qui, pour rappel, est constituée des deux opérations

pour i allant de 0 à 255 :

j=(j+Si+Ki) modulo 256 ; Echanger(Si,Sj)

donnera le résultat suivant pour les trois premières itérations :

Initialisation i=0

- j= (0+0+3) = 3
- S =

i	0	1	2	3	4	5	…	…	…	255
Si	3	1	2	0	4	5	…	…	…	255

Initialisation i=1

- j= (3+1+255) modulo 256 = 3
- S =

i	0	1	2	3	4	5	….		…	255
Si	3	0	2	1	4	5	…		…	255

Initialisation i=2

- j= (3+2+V) modulo 256 = 5 + V
- S =

i	0	1	2	3	4	5	5+V	…	…	255
Si	3	0	5+V	1	4	5	2	…	…	255

Initialisation i=3

- j= (5+V3+1+K_3) modulo 256 = 6+V+K_3
- S =

i	0	1	2	3	4	5	5+V	6+V+K_3	…	255
Si	3	0	5+V	6+V+K_3	4	5	2	1	…	255

Génération du 1er Octet de flux chiffrant

Pour rappel, l'étape de génération est la suivante :

i=(i+1) modulo 256 ;
j=(j+Si) modulo 256 ;
Echanger(Si,Sj) ;
Sélectionner $S_{(Si+Sj) \text{modulo } 256.}$

Nous obtenons alors :

- i = 1
- j = (0)
- S

i	0	1	2	3	4	5	5+V	6+V+K_3	...	255
Si	0	3	5+V	6+V+K_3	4	5	2	1	...	255

- Octet chiffrant = $S_{(0+3)}$= 6 + V + K_3

Le cryptogramme est obtenu au moyen de l'opération XOR entre le texte plein M et le flux chiffrant K

$$C = M \text{ xor } K$$

Ceci implique que :

$$C \text{ xor } M = K$$

Si nous appliquons cela à l'exemple ci-dessus, nous pouvons écrire :

$$C \text{ xor } M = 6 + V + K_3$$
$$\mathbf{K_3 = C \text{ xor } M - 6 - V}$$

Sur base de l'exemple ci-dessus, on constate qu'au moyen d'un vecteur d'initialisation de la forme [3,255,V], il est possible de retrouver le quatrième octet de la clé de chiffrement (les trois premiers étant le vecteur d'initialisation lui-même).

Cette affirmation comporte certaines restrictions :

Premièrement, le premier octet du message en clair doit être connu. Les trames 802.3 sont les données utiles encapsulées dans les trames 802.11. Il s'avère que ces trames 802.3 contiennent des en-têtes (LLC/SNAP) qu'il est possible de déterminer [Fluhrer, 2001]. Il est donc possible de connaître le premier octet de texte en clair. Nous sommes par conséquent dans le cas d'une attaque à clair connu.

Secondement, l'exemple ci-dessus arrête la phase « KSA » après trois itérations, or elle en contient 256. Afin de pouvoir retrouver un octet de la clé il faut s'assurer que durant les 253 itérations restantes, le premier, le deuxième et le quatrième octet de S ne seront pas modifiés.

Cette probabilité est évaluée à 5% d'après [M.LOW, 2007]. En effet:

- L'indice « i » est incrémenté à chaque itération, et n'influencera plus les 4 premiers octets.

- Nous pouvons considérer que l'indice J a 253 chances sur 256 d'être strictement plus grand que 3 à chaque itération. Etant donné qu'il reste 252 itérations nous pouvons calculer la probabilité P d'avoir les quatre premiers octets inchangés de la façons suivante :

$$P(253/256)^{252}=0,0513$$ [M.LOW, 2007].

Cette probabilité est ensuite utilisée pour déterminer le nombre de trames ayant un vecteur d'initialisation de la forme [3,255,V] à intercepter. En appliquant la méthode définie ci dessus nous retrouverons dans 95% des cas un octet aléatoire, et dans 5% un octet de la clé. En appliquant cette méthode un nombre suffisant de fois, l'octet de la clé aura une fréquence d'apparition plus importante que les octets aléatoires.

Finalement il reste à appliquer cette méthode aux octets suivants.

Ainsi, pour obtenir le **cinquième** octet de la clé K nous utiliserons un vecteur d'initialisation de la forme:

Tableau 4: IV

i	0	1	2
Si	**4**	255	V

En appliquant le même algorithme nous obtenons la table d'état S suivante après la **quatrième** itération :

Initialisation i=4

- $j= (9+V+1+K_3)+1+K_4$ modulo 256 = $10 + V + K_3 + K_4$
- S =

i	0	1	2	3	4	5	6+V	9+V+K_3	10+V+K_3+K_4	255
Si	4	0	6+V	9+V+K_3	10+V+K_3+K_4	5	2	3	1	255

Génération du 1er Octet de flux chiffrant

- i = 1
- j = (0)
- S

i	0	1	2	3	4	5	6+V	9+V+K_3	10+V+K_3+K_4	255
Si	4	0	6+V	9+V+K_3	10+V+K_3+K_4	5	2	3	1	255

- Octet chiffrant = $S_{(0+4)}$= 10+V+K_3+K_4.
  K_3 ayant été déterminé à l'étape précédente, il est possible de calculer K_4.

3.1.2.1.1 Exemple

Afin d'illustrer cette attaque, prenons le cas simplifié suivant constitué:

- d'un message M à transmettre tel que : M = [8]» = [0 0 0 0 1 0 0 0]
- d'une clé secrète K = [3]
- d'un vecteur d'initialisation IV = [3 ;255 ;2] soit de la forme [3 ;255 ;V]

Le message chiffré C calculé par RC4 est :
Initialisation i=3

- j= (5+V+1+K_3) modulo 256 = 6+V+K_3= 8 + K_3 (avec K_3=3) et (V =2) = 11
- S =

i	0	1	2	3	4	5	7	11	…	255
Si	3	0	7	11	4	5	2	1	…	255

Lors de la génération,l'octet chiffrant sélectionné (F) est S_3 soit 11.

Nous pouvons alors calculer C tel que :

$$C = M \text{ xor } F \text{ (soit 8 xor 11).}$$

Ce qui donne :

	M	0 0 0 0 1 0 0 0]
xor	F	0 0 0 0 1 0 1 1
	C	0 0 0 0 0 0 1 1

Ayant repéré la forme du vecteur, l'attaquant sait que $F = 6 + V + K_3$ et implicitement que :

$$K_3 = F - 6 - V \text{ soit } K_3 = F - 8$$

Le première octet du message en clair est supposé connu et permet de déterminer F tel que

$$F = M \text{ xor } C$$

Soit

	M	0 0 0 0 1 0 0 0
xor	C	0 0 0 0 0 0 1 1
	F	0 0 0 0 1 0 1 1

Nous avons finalement :

$$K_3 = 11 - 8 = 3$$

Conclusion

En conclusion, cette attaque est basée sur une classe de vecteurs d'initialisation dits « faibles » de la forme [n,255, V] et permettant de retrouver l'octet « n+1 » de la clé secrète. Il est possible de déterminer d'autres vecteurs utiles en exécutant les « n » premières étapes de l'algorithme « KSA ». En pratique cette attaque nécessite de récolter 500.000 trames.[Beck, 2008]

Par la suite, de nouvelle classes de vecteurs faibles on été découvertes notamment via l'attaque de KoreK [Bittau, 2006] réduisant ainsi le nombre de trames à capturer.

Afin d'améliorer le temps d'exécution de cette attaque, nous pouvons accroître le trafic en injectant artificiellement des requêtes ARP. [Fluhrer, 2001]

Chopchop

L' attaque chopchop a été proposée par Korek et est basée sur l'utilisation inappropriée du CRC32 dans WEP. En effet le protocole WEP utilise CRC32 afin de protéger l'intégrité des trames contre une modification volontaire. Or, CRC32 a été conçu pour détecter des erreurs de transmission involontaires. Son caractère linéaire le rend prédictif, il permet de modifier une trame cryptée tout en maintenant un code CRC32 correct. [Bittau, 2006]

Sur base d'un cryptogramme C intercepté, il est possible de calculer un nouveau cryptogramme C' tel que

$$C' = C \text{ xor } (M', CRC32(M'))$$ [Borisov, 2002]

« C' » etant valide, et « M' » étant un message connu de l'attaquant.

Sur base de cette propriété, l'attaque chopchop permet de décrypter n'importe quelle trame sans connaître la clé de chiffrement.

La technique consiste à supprimer le dernier octet d'un cryptogramme. Le message devient donc corrompu. Cependant, si on applique l'opération xor au messsage obtenu avec une valeur particulière, le cryptogramme redeviendra valide. Cette valeur correspond à l'octet supprimé.[Borisov, 2002]

Il ne reste donc qu'à essayer successivement les 256 valeurs possibles et à retransmettre la trame ainsi obtenue vers le point d'accès. Si ce dernier rejette la trame, l'octet choisi n'est pas correct, sinon c'est le bon.

Cette attaque est possible car WEP ne propose aucune protection contre le rejeu.

Attaque par Fragmentation (Andrea Bittau)

L'attaque par fragmentation est décrite ci-dessous sur base d'un article [Bittau, 2006] publié par son auteur.

Nous avons vu dans le chiffrement de Vernam qu'il ne faut jamais utiliser deux fois la même clé lors du chiffrement. L'espace de clé du protocole WEP est réduit à 2^{24} possibilités. Ce nombre de trames est atteint en moins d'une journée sur un réseau relativement actif [Fluhrer, 2001], d'autant plus que la norme 802.11 n'oblige pas à modifier le vecteur d'initialisation pour chaque trame.

Si l'attaquant parvient à obtenir un message en clair et son correspondant chiffré (ce qui est notamment possible lorsque l'authentification en mode clé partagée est utilisée), il possèdera le flux d'octets chiffrants associé au vecteur d'initialisation de la trame. Par conséquent l'attaquant aura la possibilité :

1. De décrypter toutes les trames utilisant le même vecteur d'initialisation [1].
2. D'émettre des trames en utilisant ce vecteur d'initialisation.

En collectant un grand nombre de ces flux chiffrants, il pourra petit à petit constituer un dictionnaire faisant correspondre pour chaque IV le flux chiffrant associé.

Emettre 64 octets

Afin de récolter plus rapidement les flux chiffrants associés aux différents IV, il est possible d'exploiter la fragmentation 802.11. En effet, la norme 802.11 permet de fragmenter chaque trame en maximum 16 trames chiffrées indépendamment.

Il faut savoir que les huit premiers octets de texte clair de chaque trame peuvent être facilement déterminés (LLC/SNAP). On peut ainsi en déduire les huit premiers octets de flux chiffrant de chaque trame. Ces huit octets permettent de chiffrer:

a. 4 Octet de données utiles.
b. 4 Octets de CRC32

En fragmentant une trame en 16 sous-trames utilisant le même vecteur d'initialisation, il est alors possible d'émettre 16 x 4 octets de données utiles, soit 64 octets.

[1] Avec une restriction sur la taille du message qui ne peut excéder la taille du flux chiffrant .

A ce stade, la fragmentation permet donc d'émettre 64 octets de données arbitraires. Nous allons maintenant voir comment l'utiliser pour récupérer des trames 1500 octets. (Taille maximum des données en provenance de la couche IP).

Récupérer des flux chiffrant de 1500 octets

La figure ci-dessous illustre un mécanisme permettant de récupérer un flux chiffrant d'une taille de 1500 octets après avoir émis 34 trames en broadcast.

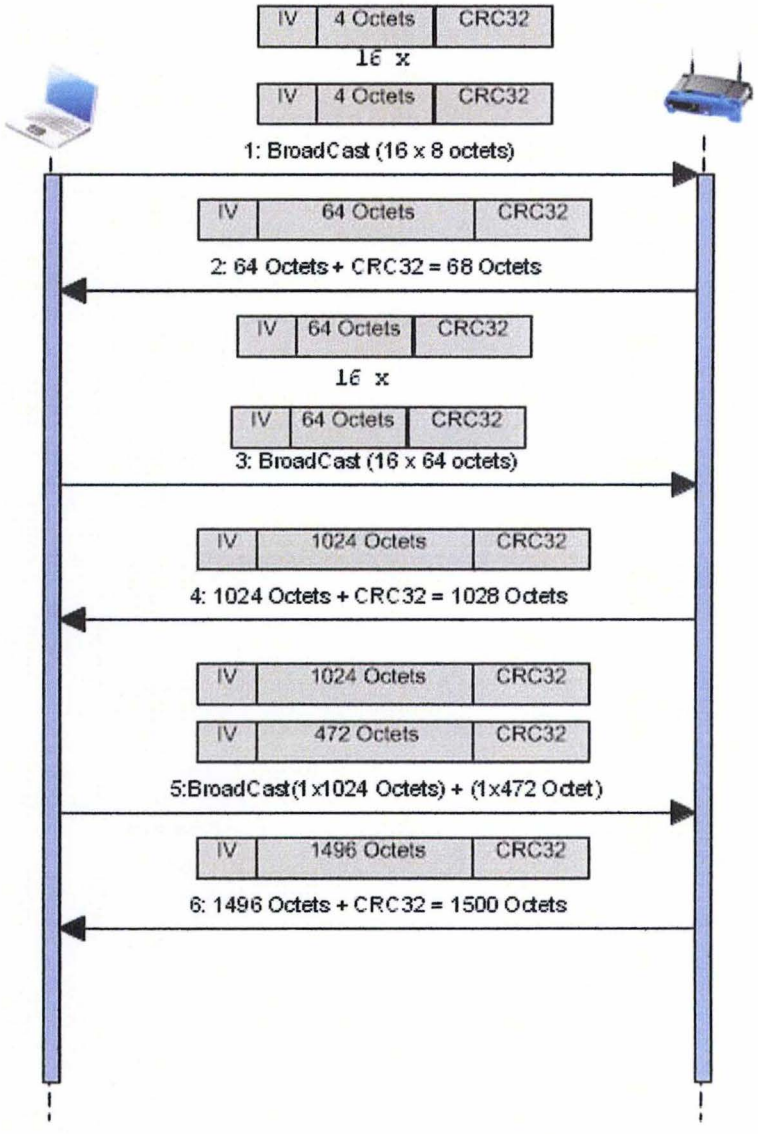

Figure 21: Fragmentation

1. Envoyer en broadcast une trame de 64 octets de données connues fragmentées en 16 trames sur base de la technique vue au point précédent.

2. Le point d'accès va réassembler la trame avant de la diffuser.

3. La station intercepte la trame reconstituée. Connaissant les données en clair, et le cryptogramme nous récupérons le flux chiffrant (K = M xor C). A ce stade nous avons récupéré un flux chiffrant de 68 octets permettant de chiffrer 64 octets de données utiles. Nous allons utiliser ce flux chiffrant afin de chiffrer un message de 1024 octets répartis dans 16 trames que nous envoyons en broadcast.

4. Le point d'accès réassemble la trame avant de la diffuser.

5. De façon similaire au point 3, nous sommes maintenant en possession d'un flux chiffrant de 1028 octets. Nous l'utilisons pour émettre 1496 octets de données sous la forme de deux trames.

6. Le point d'accès reconstruit la trame avant de la diffuser. La station qui l'intercepte est alors en possession d'un flux chiffrant de 1500 octets. Ce flux chiffrant ainsi que le vecteur d'initialisation sont stockés dans un dictionnaire afin de pouvoir émettre des trames de longueur maximum, et de décrypter toutes les trames interceptées qui utilisent ce vecteur d'initialisation.

Décrypter les trames en temps réel

Pour le décryptage, la fragmentation peut être utilisée afin de d'ajouter un en-tête IP à une trame capturée avant de la rejouer. Lorsque le point d'accès recevra cette trame il la déchiffrera et la transmettra notamment à l'hôte se trouvant à l'adresse IP spécifiée par l'attaquant.

Cette technique se déroule de la façon suivante:

1. Capturer une trame.

2. Préparer les fragments nécessaires pour représenter un en-tête IP , au moyen d'un flux chiffrant récupéré avec l'une des méthodes mentionnées dans les points précédents.

3. Emettre les fragments IP et la trame capturée vers le point d'accès.

4. A la réception le point d'accès réassemble la trame et la transmet aux adresses IP spécifiées (notamment celle ajoutée par l'attaquant). Ceci est possible grâce au fait que chaque fragment est chiffré indépendamment. Le contrôle d'intégrité est effectué sur chacun d'eux, mais aucun contrôle n'est effectué sur l'ensemble de la trame reconstituée.

Si l'ajout de ce fragment entraine un dépasse 1500 octets, il est possible, au moyen de la technique chopchop, de modifier l'en-tête IP déjà présent au lieu de l'ajouter.

Cette attaque nécessite néanmoins de connaître l'adresse MAC du routeur et une adresse IP valide.

Conclusion

D'autres attaques continuent à être publiées, notamment une attaque proposée par Martin Beck en novembre 2008 qui permet de retrouver les flux chiffrants sans utiliser de vecteurs faibles.

Nous pouvons résumer les faiblesses du WEP de la façon suivante :

- Une mauvaise gestion de l'intégrité par l'algorithme CRC32.
- Aucune protection contre le rejeu.
- Un espace de clé trop limité.

De nombreuses attaques ont été publiées prouvant définitivement que le WEP ne peut plus être considéré comme un algorithme cryptographiquement sûr.

3.2 WPA - TKIP

3.2.1 Présentation

Considérant les failles du protocole WEP, le WPA (Wi-Fi Protected Access) a été créé en 2003 par la Wi-Fi Alliance comme une solution intermédiaire offrant une compatibilité avec le matériel existant. Afin d'assurer cette compatibilité, le protocole WPA continue à utiliser l'algorithme RC4 mais sa mise en œuvre a été révisée. En effet, les fonctions logiques de chiffrement RC4 sont souvent imprimées dans les circuits électroniques des adaptateurs. Afin de pouvoir maintenir l'utilisation des adaptateurs existants, il était primordial de proposer une solution intermédiaire basée sur RC4. WPA, répond à cette contrainte par la mise en œuvre d'un mécanisme permettant de renouveler la clé WEP pour chaque trame au lieu d'utiliser une clé unique. Ce mécanisme de chiffrement est appelé TKIP (Temporal Key Integrity Protocol) et est présenté dans la figure de la page suivante.

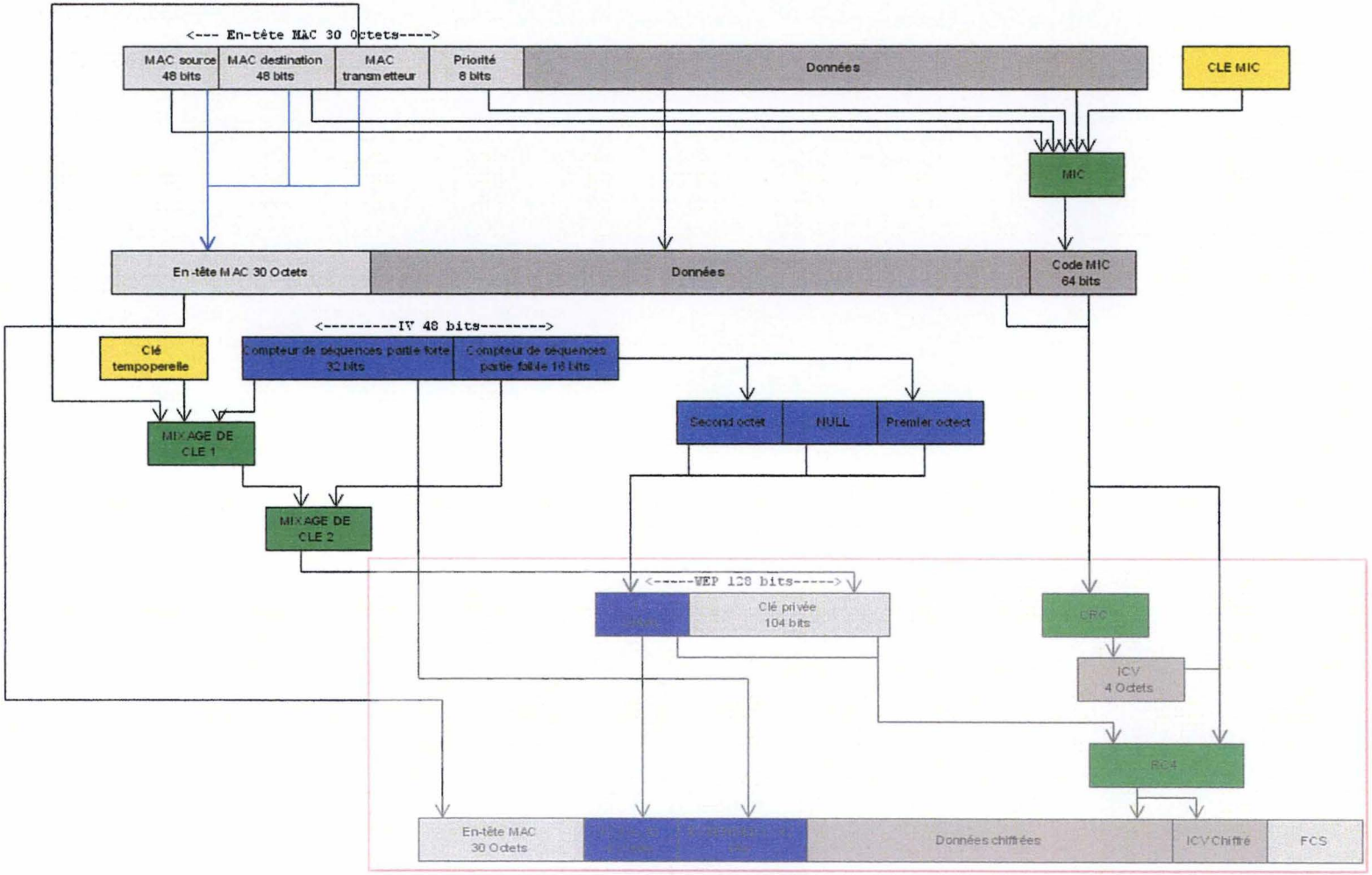

Figure 22: TKIP

Dans la figure ci-dessus (basée sur une figure de [Gast, 2005]), la zone grisée délimitée par le cadre rouge correspond au WEP et met en évidence que le fait que le protocole WPA consiste uniquement à «préparer » les paramètres d'entrée du protocole WEP.

Par rapport au WEP, WPA apporte les améliorations suivantes :

1. Le vecteur d'initialisation est désormais sur 48 bits au lieu de 24 bits, ce qui est suffisant pour empêcher la réutilisation d'un même flux de chiffrement.

2. Les vecteurs d'initialisation qui produisent des clés faibles sont écartés.

3. La clé de chiffrement n'est plus fixe mais change dynamiquement au moyen du protocole TKIP.

4. Le contrôle d'intégrité est assuré au moyen du protocole MIC (cf. Contrôle d'intégrité MIC (Michael))

5. Un compteur est ajouté à chaque trame pour empêcher le rejeu.

Sur la « Figure 22:TKIP » nous pouvons constater que TKIP fait usage de différentes clés (MIC et temporelle en jaune). Ces clés sont dérivées d'une clé maître (PMK) obtenue après le processus d'authentification qui peut être soit :

1. PSK (Pre-Shared Key) : une clé secrète est préalablement enregistrée sur le point d'accès et sur les clients. Généralement cette clé de 256 bits est générée sur la base d'une phrase de passe allant de 8 à 63 caractères. Le WPA qui repose sur cette méthode d'authentification est aussi appelé WPA-PERSONNAL.

2. 802.1X : une méthode « EAP »(cf. 4.4.4) permet l'authentification de l'utilisateur et la distribution des clés [Atelin, 2008]. Si lors de ce processus le client est authentifié, une clé de 256 bits est calculée. Cette méthode est aussi appelée WPA-ENTREPRISE.

La figure suivante extraite de la « Figure 8: Etapes du dialogue 802.11 » met en évidence (cadre rouge) le moment où la clé maître **(PMK)** est générée.

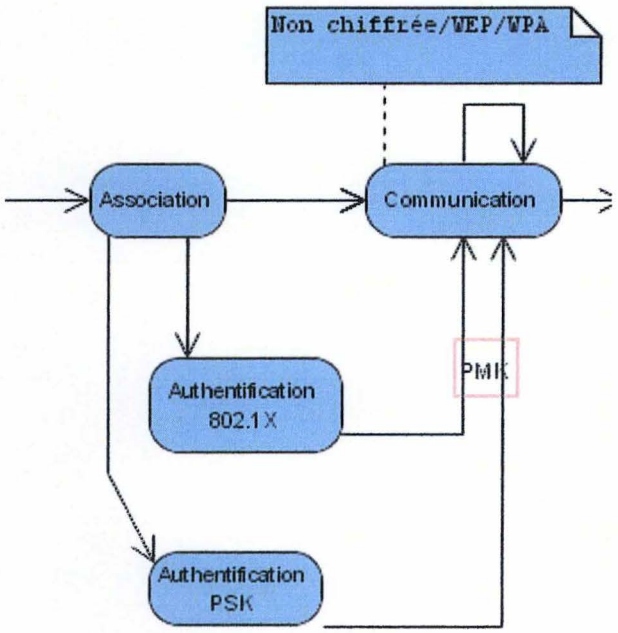

Figure 23: PMK

Quel que soit le type d'authentification utilisé, lorsqu'une station cliente est authentifiée une clé de 256 bits est mise à disposition. Cette clé est appelée PMK (pairwise master key) et sera utilisée comme base pour la génération des autres clés nécessaires au chiffrement et au contrôle d'intégrité des trames. Ce processus de dérivation des clés est défini dans « La hiérarchie de clés 802.11»[Gast, 2005].

Hiérarchie des clés

Comme spécifié au point précédent, la norme 802.11 définit une hiérarchie de clés. Celle-ci va permettre de générer deux types de clés:

1. Les clés temporelles et les clés MIC (cf. Figure 22:) qui changent pour chaque trame.

2. Les clés utilisées pour permettre l'échange des clés temporelles.

Cette hiérarchie dépendra :

1. Du mode de diffusion (Unicast, multicast) utilisé.

2. De la méthode de chiffrement utilisée TKIP/CCMP.

<u>En mode Unicast et pour TKIP</u>

L'utilisation d'une hiérarchie de clés, dans le cadre d'une communication en mode unicast, permet de changer de clés de chiffrement sans répéter le processus d'authentification [Gast, 2005]. Cette hiérarchie est constituée des clés suivantes :

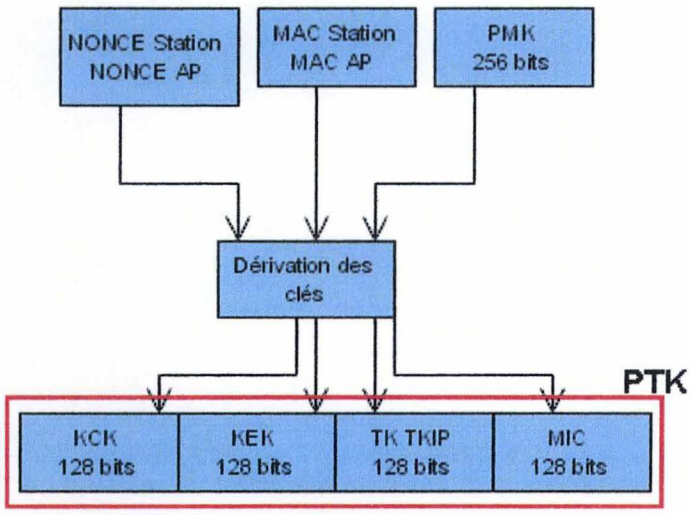

Figure 24: Clé UNICAST

1. **Pairwise Master Key (PMK)** : aussi appelée la « clé maître », cette clé de 256 bits, dépend directement du mode d'authentification - comme illustré à la figure « Figure 23: PMK »- . Cette clé n'est jamais utilisée pour le chiffrement mais bien pour générer d'autres clés temporaires.

2. **Pairwise Transient Key (PTK)** : cette clé est calculée pour chaque trame sur base de la clé maître, des adresses MAC de l'émetteur et du récepteur, et de deux numéros aléatoires. Dans le cadre du protocole (TKIP)[1], elle est de 512 bits et est calculée en utilisant une fonction de hachage SHA1 (cf. Annexe 1) implémentant la RFC 2104 concernant HMAC(cf. Annexe 2). Les 512 bits de la PTK constituent 4 clés de 128 bits :

    1. **Key Confirmation Key (KCK)** : utilisée dans le dialogue 802.1X (cf. 4.1).

    2. **Key Encryption Key (KEK)** : utilisée dans le dialogue 802.1X (cf. 4.1) notamment pour chiffrer l'échange des nouvelles clés temporelles.

    3. **Temporal Key (TK)** : utilisée pour le chiffrement de la trame ; illustré en jaune sur le figure (cf. Figure 22: TKIP)

    4. **Temporal Mic Key (TMK)** : utilisée pour le contrôle d'intégrité ; illustré en jaune sur le figure (cf. Figure 22: TKIP)

En mode multicast :

1. **Group Master Key (GMK)** : cette clé correspond à la clé « maître » dans une transmission unicast. N'étant pas fournie par le processus d'authentification, elle est calculée de façon aléatoire et a une taille de 128 bits.

3. **Group Temporal Key (GTK)** : cette clé est calculée à partir d'une fonction aléatoire prenant en entrée la clé GMK, l'adresse MAC du point d'accès, et un nombre aléatoire. Elle est renouvelée lorsqu'un client quitte le réseau [Atelin, 2008] et est constituée (dans le cadre du protocole TKIP) de deux clés de 128 bits :

    1. **Group Encryption Key (GEK)** : utilisée pour le chiffrement des données.

    2. **Goup Integrity Key (GIK)** : utilisée pour le contrôle d'intégrité.

[1] Cette clé est différente dans le cadre du protocole CCMP (cf. 3.3.1).

Renouvellement et distribution des clés

Afin de générer les clés temporaires PTK et GTK, un protocole de « négociation en quatre phases » ou « 4-Way HandShake » est appliqué. Ce protocole est décrit dans la figure ci-dessous.

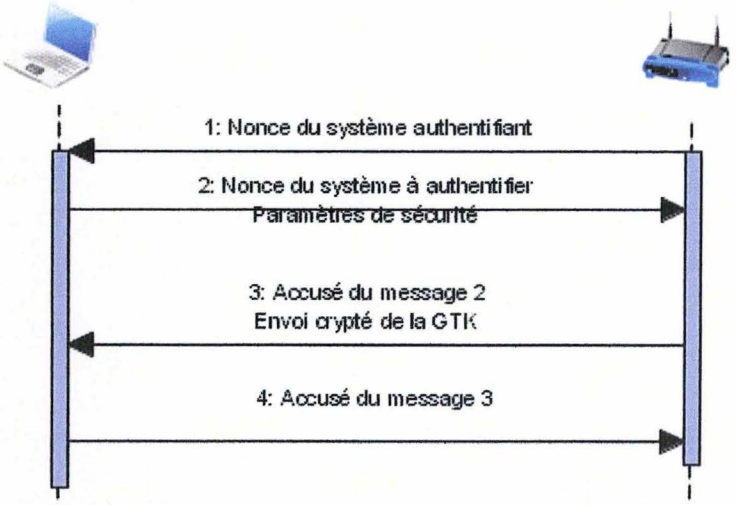

Figure 25: Négociation en 4 phases [Gast, 2005]

1. Le système authentifiant initie la négociation par l'envoi d'une valeur aléatoire « nonce » au système à authentifier. Ce dernier génère un second nonce et la hiérarchie de clés sur base des adresses MAC, de la PMK, et des deux nonces. (cf. Figure 26 : Génération hiérarchie des clés partie A) [Lehembre, 2006].

2. Le second nonce, signé au moyen de la clé KCK, est envoyé au système authentifiant (cf. Figure 26 : Génération hiérarchie des clés partie B).
   A la réception, le système authentifiant calcule à son tour la hiérarchie de clés sur base du nonce reçu et vérifie la signature. La négociation s'arrête si la vérification échoue. (cf. Figure 26 : Génération hiérarchie des clés partie C)

3. Le système authentifiant envoie alors la GTK chiffrée au moyen de la KEK signée avec la KCK. (cf. Figure 26 : Génération hiérarchie des clés partie D). A la réception, le système à authentifier prend connaissance de la GTK et vérifie la signature.

4. Le dernier message permet de confirmer au système authentifiant que toutes les clés sont en place et que la communication chiffrée peut commencer. Ce dernier message est à nouveau signé sur base de la KCK (cf. Figure 26 : Génération hiérarchie des clés partie E).

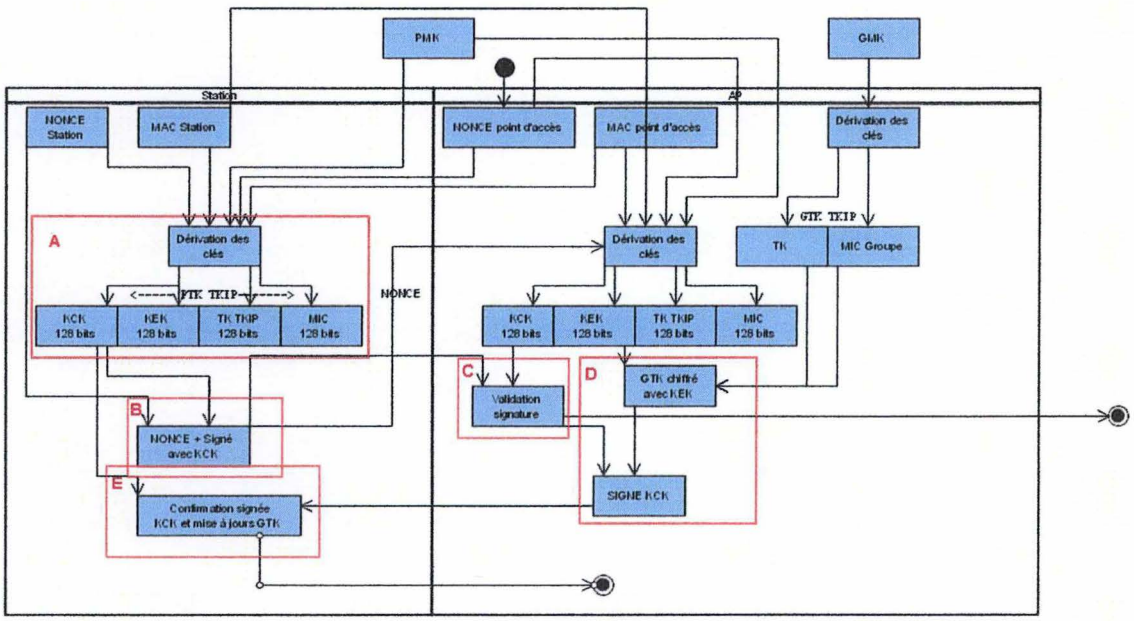

Figure 26 : Génération hiérarchie des clés

Au moyen de la procédure de négociation en quatre phases décrite ci-dessus, la GTK a été calculée. Afin de renouveler la GTK, une procédure de négociation de clé de groupe spécifique est utilisée et se déroule en deux phases :

Figure 27: Calcul GTK

1. La GTK est générée, chiffrée au moyen de la KEK, authentifiée au moyen de la KCK et envoyée au système à authentifier par le système authentifiant.

2. Un accusé de réception authentifié par la KCK est envoyé au système authentifiant et confirme l'utilisation de la nouvelle GTK.

Cette procédure a lieu lorsqu'une station s'associe ou se désassocie du point d'accès.

Contrôle d'intégrité MIC (Michael)

Comme mentionné au début de ce chapitre, le protocole WPA assure l'intégrité des trames chiffrées grâce au contrôle d'intégrité MIC.
Celui-ci a été développé afin de réduire les risques liés à l'utilisation d'un contrôle d'intégrité linéaire tel que CRC 32, tout en veillant à ne travailler qu'avec des opérations « simples » (XOR, décalage, addition) afin de garantir une compatibilité avec l'ancien matériel conçu pour le WEP.

Le contrôle d'intégrité MIC génère un code MIC de 64 bits sur base des paramètres suivants :

1. TMK.
2. L'adresse MAC du destinataire.
3. L'adresse MAC de l'émetteur.
4. La trame non fragmentée en clair.

Le code MIC ainsi obtenu est ajouté à la fin de la partie « données » de la trame. Comme le code MIC est calculé sur la trame non fragmentée il faut réceptionner tous les fragments avant de pouvoir le vérifier.

Le contrôle d'intégrité Michael n'étant pas infaillible, certaines contre-mesures sont définies dans la norme 802.11. Si une erreur dans le code MIC est détectée, elle est notifiée. Si une seconde erreur est détectée dans un intervalle de temps de 60 secondes, les communications sont arrêtées durant 60 secondes et le processus de renouvellement des clés est lancé sur toutes les stations.

En effet, on considère qu'une erreur au niveau du code MIC doit être exceptionnelle car pour arriver à ce stade il a fallu préalablement passer la protection contre le rejeu et le contrôle d'intégrité WEP de chaque fragment. [Gast, 2005]

Interaction des composants (Temporal Key Integrity Protocol)

Le protocole TKIP, permet la compatibilité avec le WEP, et peut être considéré comme une fonction dont la sortie est une clé RC4 de 128 bits servant d'entrée au protocole WEP. Le fonctionnement détaillé du protocole TKIP est illustré à la figure « Figure 22: TKIP » au début de ce chapitre.

Le point ci-dessous décrit les opérations effectuées sur une trame 802.11 lors de l'émission. Ces étapes sont détaillées sur base d'informations provenant de [Gast, 2005] et [Atelin, 2008].
Comme pour le WEP, seules les données utiles sont chiffrées tandis que les entêtes de la trame restent inchangés.

Les étapes d'émission d'une trame 802.11 avec TKIP (cf. Figure 22: TKIP).

1. Le code MIC est calculé sur base de la clé temporelle, des adresses MAC destinataire et récepteur, des bits de priorités[1] et des données utiles de la trame.

2. En cas de fragmentation, chaque fragment reçoit un numéro de séquence différent. Ce numéro de séquence correspond au vecteur d'initialisation de 48 bits. (IV sur la Figure 22: TKIP). Ce vecteur d'initialisation permet de garantir la non réutilisation d'un même IV pendant la durée de vie de la clé maître. De plus, le vecteur d'initialisation se comporte comme un compteur de séquences, initialisé lors de l'initialisation de la clé maître et incrémenté de 1 lors du chiffrement de chaque trame. La valeur du compteur de séquences de chacune des stations est mémorisé, et seules les trames ayant un numéro de séquence supérieur au numéro mémorisé seront acceptées par le point d'accès.

[1] Un amendement à la norme 802.11 portant sur la qualité de service permet de définir des priorités

3. La clé RC4 utilisée par WEP est calculée de façon aléatoire au moyen d'un processus de mixage de clés en deux phases :

    1. La première phase prend en entrée les 32 bits de poids fort du vecteur d'initialisation, la clé temporaire et l'adresse MAC émetteur et génère une nouvelle clé sur 80 bits. Cette première phase a pour propriété d'avoir une sortie invariable tant que le paramètre IV est constant. Ce qui permet de n'effectuer cette opération que toutes les 2^{16} trames.

    2. La seconde phase utilise les 16 bits de poids faible du vecteur d'initialisation et la clé calculée durant la phase 1 pour produire une clé RC4 de 128 bits. Le vecteur d'initialisation sur 24 bits correspondant à cette clé est construit de la façon suivante :

        - Premier octet : le deuxième octet des 16 bits de poids faible du compteur de séquences.
        - Second octet : octet nul afin de pallier aux attaques sur les IV faibles.
        - Troisième octet : le premier octet des 16 bits de poids faible du compteur de séquence.

4. Le chiffrement RC4 est appliqué sur la trame et le code MIC à partir de la clé calculée à l'étape 3. La structure de la trame résultante est reprise dans le tableau ci-dessous mettant en évidence l'encapsulation de la trame WPA dans la trame WEP afin de garantir la compatibilité des deux protocoles.

Bit	X	32	24	...	64	8	32
WEP	En-tête MAC	IV 24 + KeyID	Données chiffrées	Données chiffrées	Données chiffrées	ICV Chiffré	FCS
WPA-TKIP	En-tête MAC	IV 24+ KeyID	IV étendu	Données chiffrées	MIC Chiffré	ICV Chiffré	FCS

Tableau 5: Comparaison des trames WEP/TKIP

3.2.2 FAILLES

Attaque par force brute

Les attaques par force brute ne sont théoriquement possibles que dans le mode PSK, le seul dans lequel un secret partagé est utilisé suffisamment longtemps pour être exploité. Cependant, de part l'espace de clé (jusqu'à 256 caractères) et le calcul assez lourd pour en dériver la PMK de 256 bits ce type d'attaque est très consommateur de temps CPU et demanderait une puissance de calcul trop importante pour pouvoir être mené dans un temps raisonnable. Il est cependant possible d'exploiter certaines phrases de passe faibles. [Gast, 2005]

Attaque de Eric Tews sur TKIP

Première attaque sur le protocole TKIP, elle a été publiée par Eric Tews, (auteur de l'attaque PTW sur le WEP) en novembre 2008.[Beck, 2008]

Comme nous l'avons vu, WPA utilise TKIP afin de générer des clés WEP différentes pour chaque trame.

Etant dans un chiffrement WEP, nous pourrions réutiliser l'attaque chopchop qui permettait de retrouver le message sans connaître la clé. Cependant cette méthode implique de rejouer un grand nombre de fois une même trame, or WPA contient une protection contre le rejeu matérialisée par le compteur de séquences.

Cependant il existe un amendement à la norme 802.11 appelé 802.11e qui porte sur la qualité de service. Cet amendement permet d'utiliser 8 canaux différents, chacun possédant son propre compteur de séquences.

Le mécanisme est alors le suivant [Beck, 2008] :

1. Capturer une trame ARP, en effet, ces trames présentent les caractéristiques suivantes:
    - Longueur constante, ce qui facilite leur identification.
    - Toujours envoyées en broadcast, (pour rappel les adresses MAC ne sont pas chiffrées) ce qui aide aussi à les repérer.
    - Une grande partie du contenu est déjà connue et seuls 15 octets sont à découvrir.

2. « Lancer une attaque chopchop modifiée afin de retrouver les octets encore inconnus ». Pour éviter de détecter le rejeu, la trame sera rejouée sur un autre canal qui contient un compteur de séquences inférieur. A ce stade nous avons deux possibilités :

    - L'octet choisi n'est pas correct, l'ICV est incorrect et la trame est écartée (le compteur n'est pas incrémenté).

    - L'octet choisi est correct, mais le code MIC est erroné. Dans ce cas il faut attendre 60 secondes pour éviter les contres mesures MIC, et ensuite l'attaquant peut tenter de déchiffrer l'octet suivant. (A nouveau le compteur n'est pas incrémenté). Il faudra donc environ 15 minutes pour décrypter la trame ARP au complet.

A ce stade nous pouvons récupérer le flux chiffrant, car nous sommes en possession du texte clair et du cryptogramme. Ce flux est toutefois valable dans le sens du point d'accès vers la station. [Beck, 2008]

3. Le MIC et le texte plein étant connus, il est alors possible de retrouver la clé MIC ayant servi à générer le code, dû au fait que le protocole MIC est, contrairement aux fonctions de hachage, réversible.

4. Au moyen du flux chiffrant et de la clé MIC, il est alors possible d'émettre une trame arbitraire uniquement sur les canaux dont le compteur de séquences est inférieur (Maximum 7 trames).

5. Les trames suivantes seront déchiffrées plus rapidement (de l'ordre de 5 minutes). Le code MIC pourra être calculé grâce à la clé récupérée.

Conclusion

Il existe à l'heure actuelle peu d'attaques connues sur le WPA.
En mode 802.1X il ne semble actuellement pas possible de casser la clé.
L'attaque de Tews ouvre cependant une première brèche dans la sécurité TKIP.

3.3 CCMP-WPA2

3.3.1 Présentation

WPA2 est une mise à jour de WPA qui est définie dans la norme 802.11i ratifiée en juin 2004 [Atelin, 2008]. Comme nous l'avons vu au point précédent, les failles du WEP étaient telles qu'il fallait développer une solution rapidement. Or la norme 802.11i n'était pas encore prête. Le WPA supportant TKIP a donc été proposé comme solution intermédiaire. La norme 802.11i définit deux types d'architectures :

1. RSN (Robust Security Netwok) : sécurité basée sur le chiffrement CCMP.

2. TSN (Transitional Security Network) : sécurité basée sur le chiffrement TKIP dans le but de préserver la compatibilité.

Nous retrouvons souvent dans la littérature (et dans les interfaces de configuration de certains contrôleurs) WPA2 associé à CCMP or, il est possible de faire du WPA2 avec TKIP.

Ce chapitre est consacré aux protocoles de chiffrement, c'est pourquoi nous parlerons de CCMP plutôt que de WPA2.

Comme illustré à la figure ci-dessous, le protocole CCMP présente les caractéristiques suivantes par rapport à TKIP:

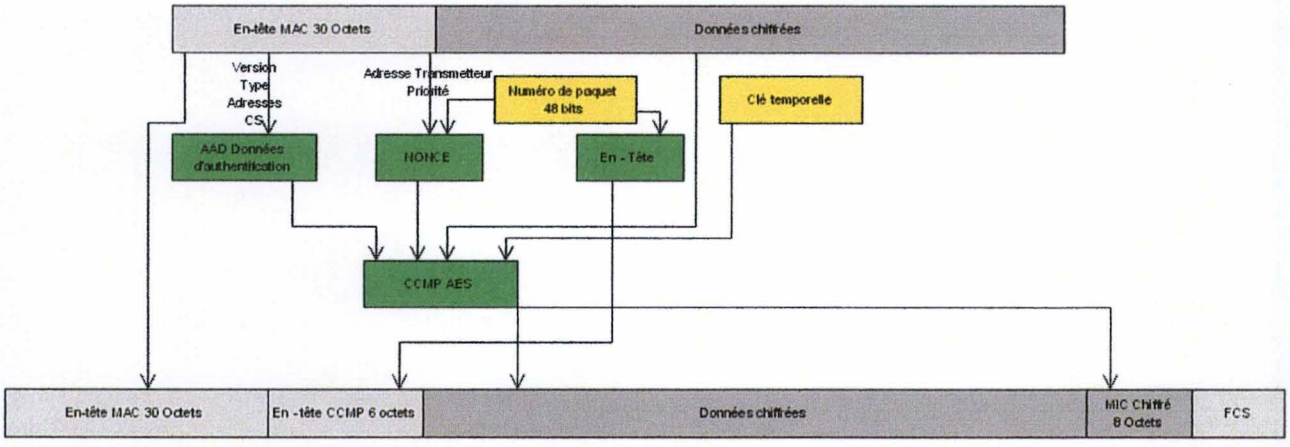

Figure 28: CCMP

1. Le chiffrement par l'algorithme RC4 est remplacé par l'algorithme AES.

2. Le contrôle d'intégrité Michael n'est plus utilisé.

3. Les deux modes d'authentification restent identiques (PSK ou 802.1X).

4. Le mécanisme de distribution des clés reste identique bien qu'au niveau de la hiérarchie des clés, les clés TMK, GIK n'existent plus, générant ainsi une PTK de 384 bits au lieu de 512 et une GTK de 128 bits au lieu de 256.

CCMP (Counter Mode with CBC-Mac Protocol)

L'algorithme CCMP, permet d'assurer l'intégrité et la confidentialité des trames en utilisant AES. (L'algorithme AES est présenté en Annexe 4)

Afin de réaliser la découpe initiale en blocs, CCMP utilise un mode de découpe particulier appelé le mode « compteur » (CTR – Counter Mode) [Swenson, 2008] qui s'approche plus d'un chiffrement par flux que par bloc. Un bloc est chiffré de façon indépendante comme dans le mode ECB, cependant un compteur est introduit afin d'éviter qu'un même bloc ne produise un même cryptogramme. Comme illustré dans la figure ci-dessous, issue de [Atelin, 2008], chaque compteur est chiffré avec la clé. Les n bits chiffrés du premier compteur sont utilisés pour chiffrer les n premiers bits de poids fort du message au moyen d'un « ou exclusif ». Tous les messages sont ainsi chiffrés au moyen du compteur correspondant.

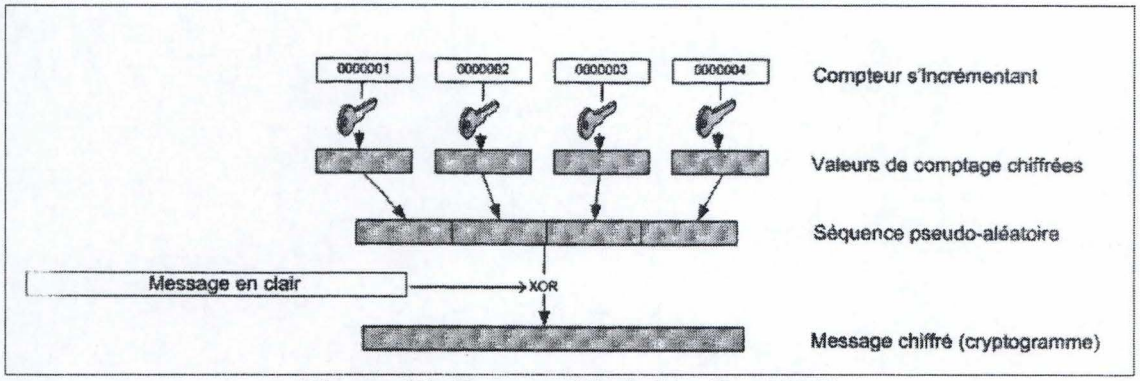

Figure 29: Counter Mode [Atelin, 2008]

Interaction des composants

Le fonctionnement détaillé du protocole CCMP est illustré à la figure suivante issue de [Lehembre, 2006].

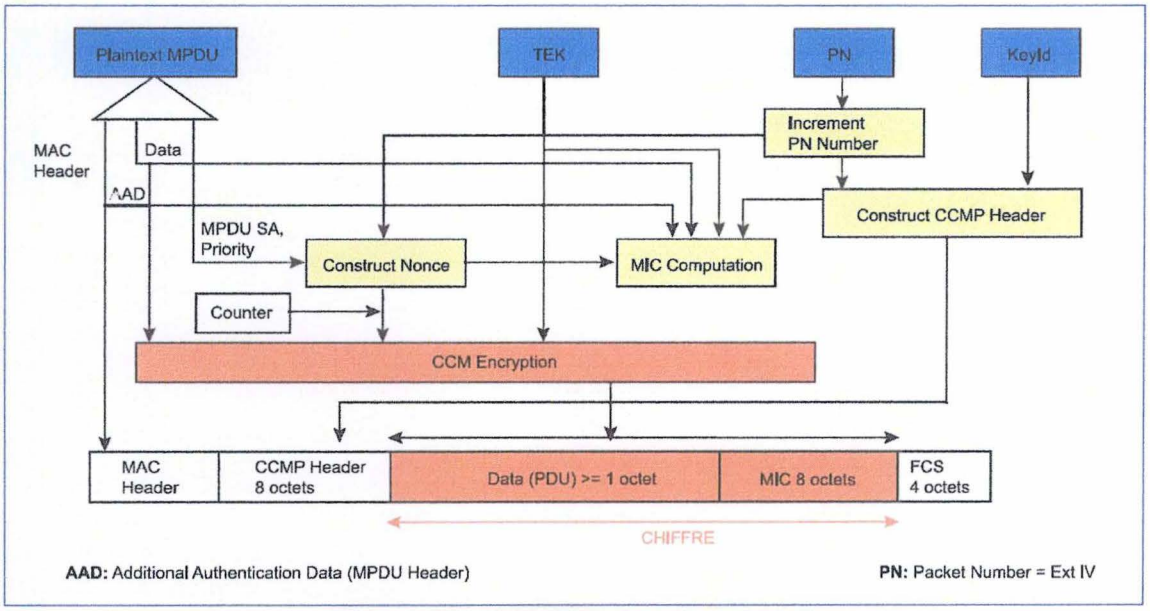

Figure 30: CCMP[Lehembre, 2006]

Dans la figure ci-dessus, les paramètres d'entrée sont illustrés en bleu :

6. **Plaintext MPDU** : la trame à chiffrer, constituée de l'en-tête MAC et des données utiles. Seules les données utiles seront chiffrées.

7. **TEK** : temporal Key (TK), cette clé est à la fois utilisée pour le chiffrement et pour l'authentification. Elle est issue du mécanisme de distribution des clés.

8. **PN** : PacketNumber, contient le numéro de paquet sur 48 bits.

9. **KeyID** : l'identifiant de la clé à utiliser .

En jaune, nous retrouvons certains traitements préalables :

1. **Increment PN Number** : permet d'incrémenter le PN lors de la transmission de chaque nouvelle trame. Ce mécanisme constitue une protection contre les attaques par rejeu.

2. **Construct Nonce** : ce processus construit le vecteur d'initialisation qui doit être unique pour une même clé sur base de l'adresse MAC de l'émetteur, du numéro de paquet, et des bits de priorité.

3. **Construct CCMP Header :** cette méthode construit un en-tête CCMP de 8 octets sur base du PN et de l'identifiant de clé. Cet en-tête contient un bit Extented IV qui est toujours fixé à 1. La structure de l'en-tête CCMP est définie dans le tableau suivant :

8 bits	8 bits	12 bits	2 bits	2bits	Octet 5	Octet 6	Octet 7	Octet 8
PN 1/6	PN 2/6	Réservé	ExtIV	KeyId	P N 3/6	PN 4/6	PN 5/6	PN 6/6

Tableau 6: En-tête CCMP

4. **MIC Computation :** cette méthode génère un code d'intégrité MIC sur 64 bits. Le contrôle d'intégrité utilisé est appelé CBC-MAC (Cipher Block Chaining with Message Authentication Code) et résulte de l'application de l'algorithme AES en mode CBC. Il utilise les paramètres suivants :

   a. L'en-tête CCMP.

   b. La clé temporaire (la même clé est utilisée pour le contrôle d'intégrité et le chiffrement).

   c. Les données utiles.

   d. Le vecteur d'initialisation.

   e. AAD (Additional Authentication Data). Ce champ permet d'authentifier des données qui doivent transiter en clair pour des raisons de compatibilité avec le protocole 802.11. Ainsi il permettra de vérifier que ces données n'ont pas été altérées.

Les traitements préalables ayant été réalisés, il reste à appliquer le chiffrement et à construire la trame CCMP.

Le chiffrement, comme cité précédemment applique l'algorithme AES en mode compteur, et prend en entrée les paramètres suivants :

1. Le vecteur d'initialisation.
2. Le texte en clair + le code MIC.
3. La clé temporaire de chiffrement.

La trame CCMP est définie par le tableau suivant :

Tableau 7: Trame CCMP

X	64 bits	...	64 bits	32 bits FCS
En-tête MAC	CCMP Headers	Données chiffrées	Code MIC chiffré	KeyId

3.3.2 FAILLES

A ce jour, aucune faille concernant le protocole CCMP n'a été publiée. Il reste cependant possible de mener une attaque par dictionnaire en mode PSK contre les mots de passe faibles.

CHAPITRE 4 : Solutions d'authentification existantes

4.1 Authentification ouverte

L'authentification ouverte se déroule avant l'association de la station au point d'accès et correspond à une authentification nulle. La station lance une requête d'authentification vers un point d'accès qui renverra toujours une réponse positive.

4.2 Authentification par clé WEP partagée

L'authentification par clé WEP partagée se déroule avant l'association de la station. Avec cette méthode, le point d'accès envoie une trame de « défi » à la station qui doit la renvoyer au point d'accès après l'avoir chiffrée au moyen d'une clé WEP partagée. Si le point d'accès peut déchiffrer la trame renvoyée, alors il considère que la station est en possession de la clé et il l'authentifie.

Lors de ce procédé, un attaquant peut facilement intercepter le texte en clair et le cryptogramme correspondant. Il peut donc en déduire le flux de chiffrement qu'il pourra réutiliser pour s'authentifier.

Cette méthode est donc déconseillée par la norme 802.11i [Gast, 2005]

4.3 Authentification par clé pré-partagée PSK(Pre-Shared Key)

L'authentification par clé pré-partagée (PSK Pre-Shared Key) a été introduite avec WPA, et se déroule après l'association de la station. Une « phrase de passe » est partagée entre la station et le point d'accès. Cette phrase est utilisée afin de dériver la clé PMK utilisée comme paramètre dans le processus de dérivation des clés (cf. Hiérarchie des clés).

4.4 Authentification 802.1X

La norme 802.1X. définit un mécanisme d'authentification qui est notamment utilisé par les protocoles WPA et WPA-2.

Trois acteurs interviennent dans ce mécanisme:

1. Le système à authentifier : la station souhaitant se connecter au réseau

2. Le système authentifiant : le point d'accès qui ne s'occupe que de l'authentification au niveau de la couche liaison. Il transmet les demandes d'authentification du système à authentifier au serveur d'authentification.

3. Le serveur d'authentification : un serveur radius qui a pour rôle d'authentifier l'utilisateur. Cette authentification peut aussi bien se faire par rapport à un annuaire LDAP, un ActiveDirectory, des domaines Windows....

Cette communication est mise en œuvre au moyen de deux protocoles différents :

- EAPOL (Extensible Authentication Protocol Over Lan) : entre le système à authentifier et le système authentifiant.

- RADIUS (Remote Authentication Dial In User Service) : entre le système authentifiant et le système authentificateur.

4.4.1 EAPOL

Le protocole EAPOL est une extension du protocole EAP.
EAP ayant pour but d'offrir un mécanisme d'authentification au niveau de la couche liaison en permettant d'utiliser différents moyens d'authentification, appelés « méthode » et travaillant sur les couches supérieures. Il ne définit pas les méthodes en tant que telles mais bien les échanges (en termes de formatage et séquencement des paquets) qu'elles nécessitent.

Un paquet EAP est défini par la structure suivante :

	8 bits	8 bits	16 bits	8bits	...
En-tête	Code	Identifiant	Longueur	Type	Données

Tableau 8: Paquet EAP

- En-tête : ce champ dépend de la couche liaison sur laquelle transite le paquet EAP, dans ce contexte : 802.11.

- Code : ce champ permet d'identifier le type du paquet qui peut être pour EAP une requête, une réponse, une notification de succès ou d'échec. EAPOL ajoute les champs suivants :
    1. EAPOL-Start : qui permet au système à authentifier de démarrer l'échange.
    2. EAPOL-Key : qui permet l'échange de clés de chiffrement.
    3. EAPOL-Logoff : qui permet de clôturer la session.
    4. EAPOL-Packet : encapsule une trame EAP.

- Identifiant : permet d'établir une correspondance entre les requêtes et les réponses

- Longueur : contient la longueur du paquet EAP, en-tête non comprise.

- Type : définit le type de la requête et/ou de la réponse :

    1. Identité : la requête de ce type ne contient pas de donnée. La réponse du même type contient généralement une forme d'identifiant de l'utilisateur.
    2. Notification : permet d'afficher un message à l'utilisateur.
    3. NAK : permet au système à authentifier de notifier qu'il ne supporte pas le système d'authentification proposé par le système authentifiant.
    4. Les autres types correspondent aux méthodes d'authentification.

- Données : ce champ contient des informations qui varient en fonction du code et du type.

4.4.2 RADIUS(Remote Authentication Dial In User Service)

Un serveur RADIUS permet de placer un intermédiaire « universel » entre le système authentifiant et un système d'authentification donné.
La structure d'un paquet radius est définie dans le tableau ci-dessous [Atelin, 2008]

	8 bits	8 bits	16 bits	128bits	...
En-tête	Code	Identifiant	Longueur	Contrôle d'intégrité	Attributs

Tableau 9: Paquet RADIUS

- Code : Ce code définit le type du paquet qui peut être :
  - Access Request : demande d'authentification
  - Access Accept : notifie le succès de l'authentification
  - Access Reject : notifie l'échec de l'authentification
  - Access-challenge : envoyé après un accès-request afin d'obtenir des informations complémentaires

- Identifiant : permet d'établir la correspondance entre les requêtes et les réponses

- Contrôle d'intégrité

- Attributs AVP (Attribute Value Pair) : notamment utilisés dans la méthode EAP-TTLS afin de transmettre des informations au serveur RADIUS. A titre d'exemple nous citons les attributs suivants : User-Name, User-Password, NAS-Port…

Le système authentifiant et le système authentificateur doivent partager une clé secrète qui sera utilisée afin de permettre l'authentification mutuelle (via un Challenge-Response). Le système authentifiant se sert ensuite de cette clé afin de contrôler l'intégrité du message en provenance du système authentificateur. Ce contrôle d'intégrité est réalisé au moyen du champ dédié contenant une signature réalisée par le système authentificateur en utilisant un algorithme de hachage prenant en entrée la donnée du paquet à émettre.

4.4.3 802.1X

Le protocole 802.1X ne définit pas de nouveaux systèmes d'authentification mais définit une méthodologie d'utilisation des méthodes d'authentification existantes, en mettant en œuvre des acteurs spécifiques communiquant entre eux au moyen de protocoles existants.

Le but de cette orchestration est de permettre un accès physique au réseau de façon indépendante du média utilisé. Cet accès physique va être matérialisé par l'association du port d'accès du système à authentifier et du port d'accès du système authentifiant.

Ces points d'accès sont nommés PAE (Port Access Entity). Le point d'accès au réseau est scindé en deux ports logiques :

1. Non contrôlé : ce port est toujours accessible mais ne permet que le trafic 802.1X.

2. Contrôlé : ce port peut être dans deux états : l'état autorisé donne accès au réseau, l'état non autorisé ne donne accès à rien. Afin de passer dans l'état autorisé, l'échange 802.1X doit avoir abouti à un succès.

Le protocole 802.1X est aussi appelé « Port based Network Protocol »

La figure ci-dessous[Gast, 2005] illustre un échange 802.1X mettant en œuvre les différents acteurs et protocoles vus précédemment.

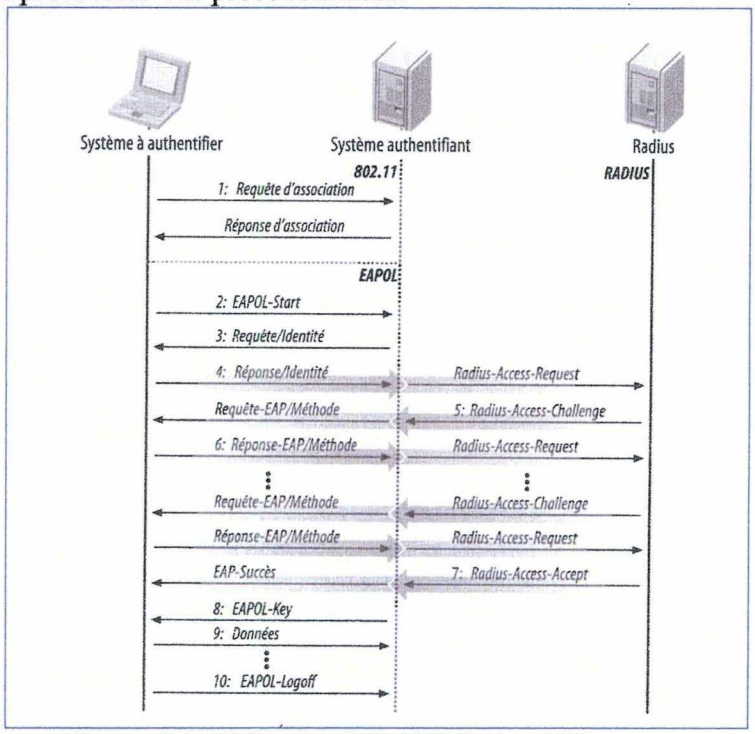

Figure 31 : Echange 802.1X[Gast, 2005]

Les commentaires suivants sont basés sur: [Gast, 2005].

1. L'association au réseau est réalisée.
2. Le système à authentifier démarre le dialogue. Le port est fermé et n'accepte que les trames EAP.
3. Le système authentifiant demande l'identité du système à authentifier.
4. La réponse d'identité est transmise au serveur RADIUS par l'intermédiaire du système authentifiant sous forme d'un paquet « Radius-Access-Request ».
5. La réponse du RADIUS « Access-Chalenge » parvient à la station sous forme d'un paquet « Request-EAP-Methode ».
6. La réponse est renvoyée au serveur RADIUS.
7. Si l'authentification aboutit à un succès, le port passe dans l'état ouvert.
8. Les clés PMK sont transmises .
9. L'échange d'informations peut se dérouler.
10. Après la communication la station émet un paquet « logoff » et les ports repassent dans l'état non autorisé.

4.4.4 Méthode EAP

Nous terminons ce chapitre par une description des méthodes EAP principales.
Une méthode EAP applicable au réseau 802.11 doit idéalement offrir les trois services suivants :

1. « Chiffrement des données identification de l'utilisateur »
2. « Authentification mutuelle »
3. « Dérivation de clés »

LEAP

Ce protocole propriétaire fait usage de MS-CHAP 1.0 et offre les trois services recommandés. Il est cependant sensible aux attaques par dictionnaire.[Gast, 2005]

EAP-TLS

Cette méthode fait usage du protocole de la couche transport TLS (Transport Layer Security) afin d'établir un tunnel sécurisé. L'authentification mutuelle est réalisée au moyen de certificats électroniques. Aussi bien le serveur que les utilisateurs doivent être en possession d'un certificat. Le client doit se référer à une autorité de certification afin de vérifier le certificat du serveur.

EAP-TTLS(Tunneled Transport Layer Security) /PEAP (Protected EAP)

Avec ces deux méthodes, seule l'authentification du serveur se fait au moyen d'un certificat. Un tunnel sécurisé est mis en place au moyen de la clé publique issue du certificat de façon similaire à TLS. Ce tunnel est ensuite utilisé afin d'appliquer un autre protocole d'authentification n'offrant pas forcément toute la sécurité nécessaire. L'établissement du tunnel est appelé « l'authentification externe », tandis que l'exécution de l'authentification au sein du tunnel est appelée authentification interne.
La méthode PEAP utilise une méthode EAP afin de réaliser l'authentification interne, tandis que la méthode TTLS utilise directement des attributs AVP (Attribute Valeur Pair).

Les méthodes EAP suivantes dont le détail sort du contexte de ce travail, n'offrent pas les trois propriétés inhérentes à un réseau 802.11 mais peuvent être utilisées comme méthodes internes avec EAP- PEAP. [Lehembre, 2006]

- MD5 Challenge : cette méthode ne génère pas de clés maître.
- GTC (Generic Token Card) : basé sur les mots de passe aléatoires et jetables.
- EAP-MSCHAP-V2.
- EAP-SIM.
- EAP-AKA.

Finalement, les méthodes suivantes, n'étant pas EAP, peuvent-être utilisées comme authentification interne avec EAP-TTLS[Lehembre, 2006]

- PAP(Password authentication protocol).
- CHAP(Challenge Handshake authentication protocol).
- MS-CHAP 1.0

CHAPITRE 5 : Analyse

Ce chapitre définit la méthode et les critères sélectionnés afin d'établir un outil d'analyse dont l'objectif sera de faciliter le choix d'une méthode d'authentification et de chiffrement lors de la mise en place d'un réseau 802.11.

5.1 Définition de la méthode

5.1.1 Justification du choix

Afin de déterminer la méthodologie à mettre en place nous avons envisagé les méthodes suivantes :

- Promethée.
- Gaia.
- GQM.
- Arbre de décision.

Les méthodes Promethée et Gaia semblent plus appropriées dans des situations où une décision doit être prise sur base de critères multiples pour lesquels les priorités sont variables. Notre cas d'étude contient des critères distincts qui vont influencer de façon plus tranchée la décision à prendre.

La méthode GQM (Goal Question Metric) est utilisée en design logiciel afin d'appliquer différents types de mesures aux processus logiciels en fonction des objectifs de l'organisation. Nous nous en inspirerons dans la mise en place de notre solution.

La méthode des arbres de décision permet d'établir une solution en procédant par élimination. Nous définissons les différentes solutions par rapport à l'objectif initial. Ensuite, nous définissons des questions en rapport avec l'objectif. Chaque réponse à une question élimine une série de solutions possibles. Nous répétons ce procédé jusqu'au moment où il ne reste qu'une solution possible. Un arbre de décision représente ce procédé : chaque nœud correspond à une question tandis que les feuilles correspondent aux solutions envisagées.

Nous avons sélectionné cette dernière méthode en nous inspirant de la méthode GQM afin de définir les questions en fonction des objectifs du réseau à mettre en place.
Le résultat obtenu sera un arbre de décision pouvant orienter les choix d'un administrateur ou d'un particulier dans le cadre de la sécurisation de son réseau Wi-Fi .

5.2 Définition des différentes solutions

Les solutions de chiffrement et d'authentification possibles sont illustrées dans le tableau suivant.

Tableau 10: Liste des solutions

Authentification	Chiffrement
Ouverte	Aucun
Clé WEP partagée	WEP
Pré-partagée WPA	TKIP-
802.1X	CCMP

Nous combinons les mécanismes d'authentification et de chiffrement afin de définir l'ensemble des solutions. Sur base des connaissances acquises dans les précédents chapitres de ce travail, nous avons pu déterminer un niveau de sécurité pour chacune des solutions afin de les ordonner.

Tableau 11: Combinaison des solutions

Authentification	Chiffrement	Sécurité
Ouverte	Aucun	Nulle
Ouverte	WEP	Faible
Clé WEP partagée	WEP	-
Clé WEP partagée	Aucun	-
Pré-partagée (WPA/WPA2)	TKIP-	Moyenne
Pré-partagée (WPA/WPA2)	CCMP-	Forte
802.1X	WEP	Moyenne
802.1X	TKIP	Moyenne
802.1X	CCMP	Forte

Suite aux failles exposées par rapport à l'authentification de type WEP, nous avons décidé d'exclure les solutions qui en font usage.

L'authentification ouverte, sans chiffrement, correspond par définition au niveau de sécurité nul.

Compte tenu des failles du protocole WEP, nous avons associé le niveau de sécurité faible à la solution utilisant le chiffrement WEP en authentification ouverte.

Nous avons respectivement considéré comme moyennes et fortes les méthodes basées sur TKIP et CMMP (sous réserve d'appliquer l'authentification 802.1X pour les structures plus importantes). Finalement, nous avons considéré comme moyenne la solution basée sur le WEP dynamique, bien qu'elle soit d'un niveau inférieure au 802.1X avec TKIP.

5.3 Définition des buts

Nous allons définir les objectifs auxquels la mise en place du réseau Wi-Fi devra répondre. Nous tenterons ensuite de déterminer les solutions de chiffrement et d'authentification les plus adéquates par rapport aux buts envisagés et offrant le niveau de sécurité le plus élevé.

Nous définissons ces buts en fonction du point de vue de deux types d'acteurs distincts :

1. Le particulier qui souhaite installer un réseau Wi-Fi à son domicile.
2. L'administrateur réseau d'une organisation.

5.3.1 Le particulier

Pour un particulier installant un réseau Wi-Fi nous avons identifié les objectifs suivants :

1. Obtenir un accès à internet.
2. Créer un réseau sécurisé entre les différentes stations de son domicile, et partager des périphériques (imprimantes, appareil photos, ...).
3. Partager sa connexion de façon sécurisée.

Un objectif inhérent à chacun de ces trois buts est la sécurité. Ainsi si nous nous référons aux principes de sécurité exposés au début de ce travail (cf. 1.3):

L'**authentification** permettra de contrôler que seules les stations autorisées peuvent accéder à son réseau et/ou à son accès internet. En effet le risque - surtout lorsque les habitations sont rapprochées - qu'un voisin utilise la connexion de façon anonyme n'est pas à négliger (avec la question légale de savoir qui est tenu responsable des actions entreprises).
L'authentification assure aussi en partie la **disponibilité**, en empêchant un utilisateur anonyme d'utiliser toute la bande passante, ou dans le cadre d'une connexion à internet d'épuiser le quota de téléchargement autorisé.

La **confidentialité** permettra d'assurer au particulier que ses données personnelles (email, photos, ...) ne transitent pas en clair dans son voisinage. L'**intégrité** lui assurera que ses données ne sont pas modifiées. Ces deux derniers principes seront assurés par les méthodes de chiffrement et de contrôle d'intégrité.

5.3.2 L'administrateur

Pour un administrateur, nous avons déterminé les buts suivants :

1. Etendre le réseau filaire existant pour les utilisateurs de l'organisation de façon sécurisée.
2. Permettre un accès sécurisé à internet et à certaines ressources à des utilisateurs extérieurs identifiables.

De façon analogue au cas du particulier, un objectif inhérent à chacun de ces buts est le respect des principes de sécurité.

Le nombre d'utilisateurs du réseau étant généralement plus important que pour le particulier et ceux-ci utilisant parfois différentes stations, l'**authentification** devra idéalement permettre d'identifier les utilisateurs plutôt que les stations. Un manque de **disponibilité** du service risquerait de paralyser les activités de l'organisation. Tandis que les questions d'**intégrité et de confidentialité** sont évidentes dans le cadre d'une organisation.

Ces buts seront atteints en tenant compte des contraintes (aussi applicables pour le particulier) suivantes :

- Transparence pour l'utilisateur : l'utilisateur final doit pouvoir profiter du réseau d'une façon simple et cohérente.
- Minimiser les coûts en faisant usage de l'infrastructure existante.

5.4 Définition des questions

Sur base des buts présentés au point précédent, et en tenant compte des contraintes, nous avons identifié des questions, dont les réponses orientent vers une solution satisfaisant un ou plusieurs buts.

La première question consiste à déterminer si nous sommes dans une organisation ou chez un particulier. Nous réserverons les méthodes d'authentification à clé pré-partagée aux particuliers. Le partage d'un secret entre les différentes stations d'une organisation ne peut pas être envisagé, pour des raisons de maintenance et de sécurité.

Le tableau ci-dessous synthétise le type de questions à mettre en évidence en fonction du type des objectifs.

Tableau 12: Questions de base

Particulier		Organisation	
BUTS	QUESTIONS	BUTS	QUESTIONS
Accès internet et réseau local	Compatibilité du matériel (WEP-TKIP-CCMP)	Extension du réseau filaire	Compatibilité des clients (WEP – TKIP – CCMP)
	Serveur RADIUS et serveur d'authentification		Infrastructure de gestion des utilisateurs existante
Partage de connexion	Serveur RADIUS et serveur d'authentification	Accès externe	Connaissance des utilisateurs externes
			Compatibilité des clients externes
			Type d'accès offert

Une question revient à tous les niveaux, à savoir la compatibilité du matériel avec les protocoles de chiffrement. En effet, le matériel utilisé déterminera le type de chiffrement que nous pouvons mettre en place. Si celui-ci ne s'avère pas suffisant par rapport aux exigences de sécurité de l'organisation, il faudra soit le remplacer, soit se diriger vers des solutions autres que le 802.11.

Au niveau de l'utilisateur particulier, la possibilité de pouvoir accéder à un serveur RADIUS pourra influencer le choix de la méthode.

Au niveau de l'organisation, la solution à préconiser sera plutôt influencée par le type de client devant se connecter, le type d'accès à offrir au monde extérieur et surtout par l'infrastructure de gestion des utilisateurs existante. Cette dernière devrait en effet influencer le choix de la méthode EAP à mettre en place.

Nous avons réalisé l'arbre de décision (page suivante) sur base de ces questions et buts.

5.5 Arbre de décision

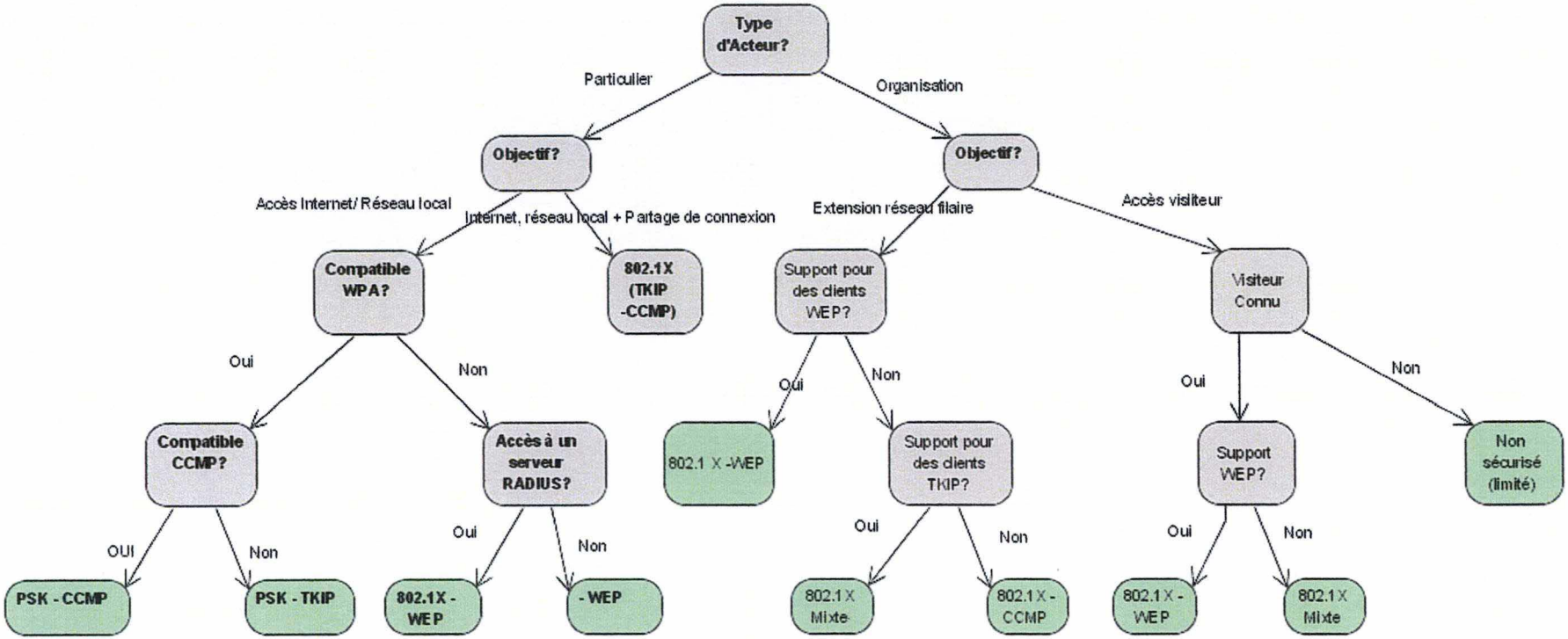

Figure 32: Arbre de décision

5.5.1 Particulier

Comme mentionné plus haut, pour une petite infrastructure ou un particulier, le mode PSK semble un choix raisonnable. En effet, il offre un niveau de sécurité suffisant, les postes étant peu nombreux, maintenir le secret partagé implique moins de travail que d'installer un serveur RADIUS dédié.

Nous orientons le choix du particulier vers la méthode la plus sécuritaire que son matériel permet. Dans l'ordre nous préconisons donc PSK-CCMP, suivi de PSK-TKIP.

Si le matériel est trop vétuste pour être compatible avec WPA, nous pouvons envisager :

- WEP statique : comme nous l'avons démontré, cette solution n'offre pas un niveau de sécurité suffisant. Il est important que le propriétaire du réseau en ait conscience. Ce système permettra au mieux de signaler que le réseau n'est pas ouvert, et empêchera les associations « accidentelles ».

- WEP dynamique : bien que théoriquement applicable, cette solution semble cependant imposante pour un particulier.

Si le particulier souhaite partager sa connexion de façon sûre tout en gérant les accès, il peut alors envisager de mettre un serveur RADIUS et un système d'authentification personnel en place. Il existe à cet effet des serveurs RADIUS open source (« FreeRadius »).

5.5.2 Organisation

Dans le cas d'une organisation souhaitant étendre son réseau, la méthode de chiffrement sera de préférence CCMP. Si les clients ne supportent pas CCMP, l'organisation pourra offrir un accès TKIP. Certains points d'accès peuvent travailler en mode mixte ce qui signifie en CCMP avec les stations qui le supportent et en TKIP avec les autres, et ce pour le trafic individuel. Pour le trafic à diffusion, le mode sélectionné correspondra à TKIP si au moins une station ne supportant pas CCMP est associée au point d'accès.

Pour les clients ne supportant pas WPA, nous pouvons envisager le WEP dynamique, ou prendre la décision de ne pas les supporter. Il est possible sur la plupart des points d'accès de supporter plusieurs SSID, laissant ainsi le choix à l'utilisateur du type de connexion qu'il souhaite utiliser en fonction de ses capacités.

Pour les accès visiteurs, nous distinguons le cas des visiteurs possédant un identifiant de celui des visiteurs inconnus. Pour un utilisateur possédant un identifiant, nous appliquerons le même processus que précédemment. L'accès au réseau sera cependant limité en fonction de la politique interne de l'organisation.
Une autre possibilité consiste à transférer la requête d'identification vers le serveur d'authentification de l'organisation du visiteur moyennant un accord préalable entre les organisations impliquées. Ce qui est par exemple le cas avec le projet « Eduroam » .

Finalement pour les utilisateurs non authentifiés il ne semble pas raisonnable d'ouvrir un accès internet. Il est cependant possible de présenter un accès à une partie de l'intranet offrant des données publiques. (Comme par exemple un moteur de recherche d'une bibliothèque).

5.6 Choix de la méthode EAP

Lorsque le mode d'authentification 802.1X a été déterminé au moyen de la méthode précédente, il reste à choisir la méthode EAP à implémenter.

Ce choix sera principalement influencé par l'infrastructure en place. Pour rappel (cf. 4.4.4 Méthode EAP) la méthode sélectionnée devra garantir un échange chiffré, une authentification mutuelle et une distribution de clés.

Les trois méthodes pouvant offrir ces mécanismes sont :

1. EAP-TTLS
2. EAP-PEAP
3. EAP-TLS

En fonction de la méthode choisie, et du système d'exploitation client, il sera parfois nécessaire d'installer un logiciel client implémentant la méthode.

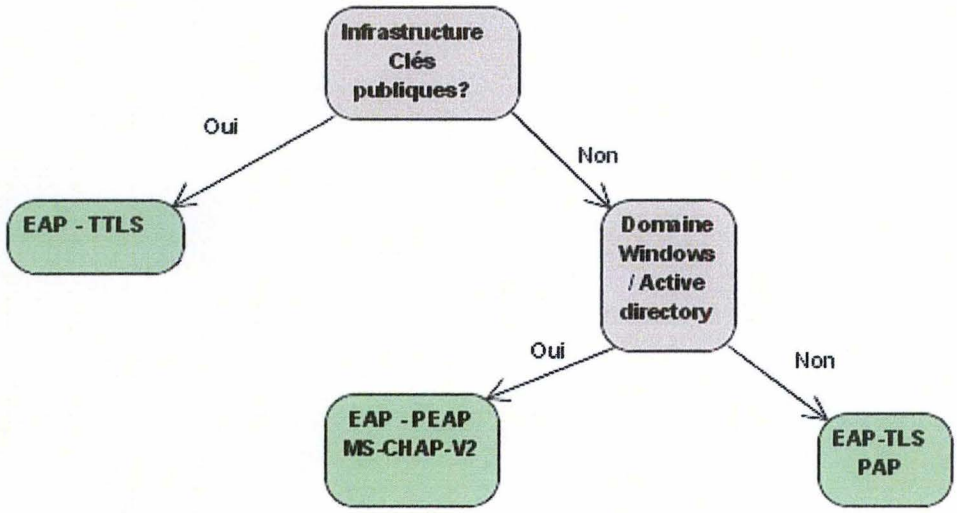

Figure 33: Arbre EAP

L'arbre de décision – figure ci-dessus – peut être lu de la façon suivante :

Si une infrastructure à clé publique est déjà en place, alors la méthode EAP-TTLS est préconisée. Cette méthode se base en effet sur les certificats client et serveur.

Lorsque la majorité des clients sont des clients Windows et que le système d'authentification des utilisateurs est ActitveDirectory ou un domaine Windows, la méthode EAP-PEAP avec Ms-CHARP-V2 est indiquée. Cette méthode est intégrée dans le système d'exploitation Windows [Gast, 2005]. Cependant, cette méthode a pour contrainte d'enregistrer le mot de passe sur la station cliente lors de la première connexion. La perte ou le vol de la station pourrait alors compromettre la sécurité du réseau.

La méthode EAP-TLS avec l'authentification interne PAP (Password Authentification Protocol) permet de faire transiter un identifiant et un mot de passe en clair (dans le tunnel TLS). Cette méthode est donc universelle et adaptée pour tout système LDAP ; UNIX, cartes à jetons. [Gast, 2005]

5.7 Intégration des paramètres non technologiques

Il existe des paramètres qui ne sont pas directement liés à la technologie mais qui pourront influencer le choix de la solution.

5.7.1 La facilité de mise en œuvre et maintenance

Au niveau d'un particulier, la facilité de mise en œuvre peut influencer le choix de la solution. Ainsi il est fréquent de voir des particuliers laisser leur réseau ouvert par facilité (et souvent par méconnaissance du risque).

De façon générale, la facilité de mise en œuvre dépendra particulièrement de la méthode d'authentification choisie : soit 802.1X, soit en clé pré-partagée. En effet la mise en place d'un serveur RADIUS, qu'il faut coupler au serveur d'authentification, est une tâche plus complexe que l'installation d'un secret partagé sur chacun des points d'accès et des postes clients et qui est exclue pour un utilisateur néophyte.

Au niveau de la maintenance le système 802.1X impliquera les tâches traditionnelles de maintenance liées à un serveur (hardware et software). Cependant il permettra une gestion plus aisée de la fluctuation des utilisateurs (ne pas changer la clé pré-partagée lorsqu'un utilisateur quitte l'organisation).

5.7.2 Le coût et la maintenance

Le coût des équipements peut influencer le choix. La solution basée sur 802.1X implique l'utilisation d'un serveur supplémentaire. Un contrôleur intégrant CCMP reste assez bon marché. Mais remplacer tous les contrôleurs WEP d'une organisation pour passer au CCMP peut s'avérer coûteux. Il peut être envisageable en fonction de la politique interne de l'organisation d'attendre le remplacement des stations.

5.7.3 La sensibilité des données

Finalement la sensibilité des données à protéger peut aussi influencer le choix de la méthode. Nous avons préconisé d'utiliser CCMP plutôt que TKIP. Nous avons aussi vu que les attaques sur TKIP sont encore limitées, impliquant une grande détermination de la part de l'attaquant. Nous pouvons prendre en considération le fait qu'un attaquant ne déploiera probablement pas de tels moyens si ce qu'il peut en retirer ne lui apporte pas une grande valeur ajoutée.

5.8 Conclusion

Nous pouvons constater que, d'un point de vue technologique, le choix d'une solution de chiffrement et d'authentification est fortement conditionné par la compatibilité du matériel, le type d'utilisateur (particulier et organisation) et l'infrastructure d'authentification existante. Nous pensons que l'arbre que nous avons développé permet de déterminer facilement la solution idéale. Toutefois, cette solution peut être reconsidérée en fonction de facteurs humains, économiques et organisationnels. Par exemple, une petite entreprise d'une dizaine de collaborateurs avec des moyens réduits et ne disposant pas des conseils d'un expert, pourrait tout à fait justifier qu'elle opte pour une solution à clé pré-partagée avec un chiffrement TKIP.

Conclusion

L'étude de la norme 802.11 nous a permis de nous créer une vue de haut niveau de l'architecture et du fonctionnement d'un réseau Wi-Fi. La compréhension de cette architecture spécifique à un réseau Wi-Fi a mis en évidence le besoin d'un mécanisme de protection des informations échangées.

La cryptographie répond à ce besoin de protection. Nous en avons donc étudié les mécanismes de base. Cette étude met notamment en évidence le besoin de chiffrement et d'authentification dans le cadre d'un échange d'informations sécurisées et permet d'appréhender leur fonctionnement dans le cadre des réseaux Wi-Fi. Afin d'étudier les faiblesses des protocoles de chiffrement et d'authentification de la norme 802.11 nous avons préalablement étudié les principaux concepts de la cryptanalyse.

Les connaissances ainsi obtenues nous ont aidés à détailler le fonctionnement des différents protocoles de chiffrement et d'authentification 802.11 ainsi que les attaques dont ils ont fait l'objet.

Après une étude approfondie de l'état de l'art et après avoir identifié les risques spécifiques aux différentes solutions, nous avons rassemblé et structuré les données nécessaires pour développer un outil destiné à simplifier le processus de décision dans le cadre de la mise en place d'une solution 802.11. A cette fin, nous avons conclu que l'arbre de décision constituait un outil adéquat. Nous avons constaté que le choix de la solution optimale en termes de sécurité était grandement influencé par la compatibilité du matériel disponible avec les protocoles récents et par l'infrastructure d'authentification existante. Nous avons alors considéré les facteurs non technologiques afin de mettre en évidence l'influence qu'ils pouvaient avoir sur la décision finale.

Nous pouvons affirmer que les protocoles de la norme 802.11 permettent de mettre en place une solution satisfaisante du point de vue de la sécurité aussi bien pour le particulier que pour l'organisation qui utilise le matériel adéquat.

Une perspective possible afin d'enrichir ce travail serait une étude des moyens de sécurisation alternatifs définis dans les couches de niveaux supérieurs et donc en dehors de la norme 802.11. En effet ces méthodes pourraient proposer des alternatives aux solutions les plus faibles et éventuellement se combiner avec les méthodes les plus fortes afin d'offrir un niveau de sécurité encore supérieur. Nous pensons par exemple à l'utilisation d'un réseau VPN qui va créer un tunnel chiffré de bout en bout. Au niveau de l'authentification, nous pensons à l'utilisation des portails captifs qui redirigent l'utilisateur vers un portail web sécurisé (HTTPS) tant qu'il n'est pas authentifié.

Références bibliographiques

[Atelin, 2008] Philippe ATELIN, *Wi-Fi – Réseaux sans fil 802.11 – Technologie, Déploiement, Sécurité,* ENI, 2008.

[Beck, 2008] Marin BECK, Erik TEWS, *« Partical attacks on WEP and WPA », 2008* (http://dl.aircrack-ng.org/breakingwepandwpa.pdf).

[Bellare, 1997] M. BELLARE, R. CANETTE, H. KRAWCZYK, « *HMAC: Keyed-Hashing for Message Authentication »,* IETF,1997 (http://www.ietf.org/rfc/rfc2104.txt).

[Bittau, 2006] Andrea BITTAU, Mark HANDLEY, Joshua LACKEY, *« The Final Nail in WEP's Coffin »*, Security and Privacy, IEEE Symposium on, 2006.

[Borisov, 2002] Nikita BORISOV, Ian GOLDBERG, David WAGNER, *« Intercepting Mobile Communications : The Insecurity of 802.11 »* ,2002.

[Buchmann, 2006] Johannes BUCHMANN, *Introduction à la cryptographie,* Dunod, 2006.

[Bryant, 2007] David BRYANT, Giovanni MOTTA, David SALOMONT, *Data Compression : the complete reference,* Springer 2007.

[Fionov, 2005] Andrey FIONOV, Boris RYABKO, *Basics of Contemporary Cryptography for IT Practitioners*, World Scientific Publishing Co. Pte. Ltd, 2005.

[Fluhrer, 2001] Scott FLUHRER, Itsik MANTIN, Adi SHAMIR, « *Weaknesses in the Key Schuduling Algorithm of RC4 »,* 2001.

[GAST, 2005] Matthew GAST, *802.11 Réseaux sans fil*, O'reilly, 2005.

[Kesley, 1996] John KELSEY, Bruce SCHNEIER, David WAGNER, « K*ey-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES* », 1996, (http://www.schneier.com/paper-key-schedule.pdf).

[Kurose, 2003] James KUROSE, Keith ROSS, *Analyse structurée des réseaux*, Pearson Education 2003.

[LEHEMBRE, 2006] Guillaume LEHEMBRE, « *Sécurité Wi-Fi – WEP, WPA et WPA2* »,hakin 9, pages 12-26,2006 (www.hsc.fr/ressources/articles/hakin9_**Wi-Fi**/hakin9_**Wi-Fi**_FR.pdf).

[Martin, 2004] Bruno MARTIN, *Codage, cryptologie et application,* Presses polytechniques et universitaires romandes ,2004 .

[M.LOW, 2007] Richard M. LOW, Mark STAMP, *Applied cryptanalysis: breaking ciphers in the real world,* Wiley, 2007.

[PICSI, 2006],Portail Internet – Cryptologie et sécurité de l'information *« Mode de chiffrement par bloc »,* 2006 (http://www.picsi.org/fiche_274.html).

[RSA], RSA Laboratories (http://www.rsa.com/rsalabs/node.asp?id=2172).

[SACCAVINI, 2003] Luc SACCAVINI, « *802.1X et la sécurisation au réseau loca l »,* 2003 (http://2003.jres.org/actes/paper.111.pdf).

[Schneier, 2001] Bruce SCHNEIER, *Cryptographie Appliquées Algorithmes, protocoles et codes source en C,*Vuibert, 2001.

[SWENSON, 2008] Christopher SWENSON, *Modern Cryptanalysis: Techniques for Advanced Code Breaking,* Wiley, 2008.

[TEWS, 2007] Erik TEWS, *« Attacks on the WEP protocol »*, 2007.

ANNEXES

1. SHA1
2. HMAC
3. CRC32
4. AES

ANNEXE 1 : SHA1

SHA1 (Secure Hash Algorithm)

Nous introduisons cet algorithme de hachage qui a été conçu par l'NIST(National Institute of Standards and Technology) et par la NSA(National Security Agency) pour être utilisé dans DSA (Digital Signature Algorithm) [Schneier, 2001].
Etant notamment utilisé dans la phase de dérivation de clé utilisée par WPA et WPA2, nous avons décidé de le détailler.

Le principe de fonctionnement de SHA1 est le suivant (repris de [Schneier, 2001]) :

1. Le message est complété (pour plus de détails cf.[Schneier, 2001]) afin qu'il ait une longueur en bits multiple de 512.

2. Initialisation de cinq variables (A-E).

3. Pour chaque bloc de 512 bits répéter :
   a. Copier les variables (A-E) dans 5 nouvelles variables (AA-EE).
   b. Pour t allant de 0 à 79.

$$
\begin{aligned}
TEMP &= (A \ll 5) + f_t(B,C,D) + E + W_t + K_t \\
E &= D \\
D &= C \\
C &= (B \ll 30) \\
B &= A \\
A &= TEMP
\end{aligned}
$$

Figure 1: SHA[Schneier, 2001]

   c. Ajouter les variables (A-E) aux variables (AA-EE) respectivement.

4. Juxtaposition des variables (AA-EE) produisant une empreinte de 160 bits.

Avec :

- K_t est une valeur fixée en fonction de l'itération sur t.

- f_t est une fonction non linéaire fixée en fonction de l'itération t :

W est un tableau de 80 mots de 32 bits. Les 16 premiers mots résultent de la découpe du bloc de 512 bits en 16. Les 75 autres sont calculés à partir des mots générés précédemment.

Annexe 2 : HMAC

HMAC (Hashed Message Authentication Code)

Cette méthode de hachage est proposée dans la RFC 2104 [Bellare, 1997] et permet de « mixer » une clé secrète dans l'empreinte générée. Afin de comprendre l'utilité d'une telle méthode, nous illustrons la problématique à partir d'un exemple issu de [M.LOW, 2007].

Présentation de la problématique

Rappelons que l'utilisation d'une fonction de hachage permet de créer une empreinte servant notamment à garantir l'intégrité du message M. Ainsi Alice enverra M + h(M). A la réception Bob calculera à nouveau l'empreinte de M et vérifiera qu'elle coïncide avec celle qui lui a été transmise. Cependant il est possible d'intercepter le message original et de le remplacer par M' + h(M'). Le destinataire ne pourra pas se rendre compte de la substitution et le calcul de l'empreinte se révèlera correct.

Afin d'éviter ce problème on juxtapose la clé secrète de Bob et Alice en fin ou en début de message dans le calcul de l'empreinte. Ainsi l'empreinte correspondra respectivement à h(K,M) ou h(M,K).

Comme nous l'avons vu pour l'algorithme SHA1, les fonctions de hachage travaillent par blocs. Pour chaque bloc du message on applique une fonction de compression « f » dont la sortie sert d'entrée à « f » du bloc suivant. Pour le bloc initial les valeurs d'entrée (IV) sont fixées. (Dans SHA1 cela correspond à l'initialisation des 5 variables).

Nous pouvons donc généraliser la fonction de hachage de la façon suivante[M. LOW, 2007] :

$h(M) = f(f(IV, M_0), M1) = f(h(M_0), M_1)$ avec M constitué de deux blocs
$h(M) = f(h(M_0, M_1 \ldots M_{n-2}), M_{n-1})$ avec M constitué de n blocs

De ceci découle :

$$h(M) = h(M')$$
$$\Leftrightarrow$$
$$h(M,X) = h(M',X) \text{ pour tout } X$$

Si M est intercepté et remplacé par M'=(M,X) :

$$h(K,M') = h(K,M,X) = f(h(K,M),X)$$

Il est dès lors possible de calculer l'empreinte du message de substitution sans connaître la clé K.
Une autre méthode se basant sur les collisions de la fonction de hachage est applicable quand la clé est ajoutée en fin message [M. LOW, 2007].

HMAC

Compte tenu de la problématique décrite au point précédent, la RFC 2104 propose une méthode permettant de « mixer » la clé au message avant d'en calculer l'empreinte.
Comme définies au point précédent les méthodes de hachage que nous étudions travaillent par blocs, posons B la longueur d'un bloc. [Bellare, 1997]

Nous définissons deux variables initiales :

1. ipad (inner pad) = B fois l'octet « 0x36 »
2. opad(outer pad) = B fois l'octet « 0x5c ».
3. h notre fonction de hachage par bloc.

Le mixage de la clé avec le message M se définit comme suit :

h(K XOR opad, h(K XOR ipad, text))[Bellare, 1997]

Annexe 3 : Exemple CRC32

L'exemple ci-dessous d'un calcul CRC est issu de [Kurose, 2003].
Sur base de :

- Message M : = [101110]
- Le polynome générateur G = [10001]
- Le degré du polynome r = 3

Nous calculons R, reste de la division de $(M * 2^r)/G$:

- $M * 2^3 = [101110000]$
- $(M * 2^3)/ G$
  - 1 0 1 1 1 0 0 0 0
    1 0 0 1
    0 0 1 0 1 0
    1 0 0 1
    0 0 1 1 0 0
    1 0 0 1
    0 1 0 1 0
    1 0 0 1
    0 0 1 1= R

Nous envoyons alors M + R soit : 101110 + 011

A la réception nous divisons alors le message reçu par G soit :

1 0 1 1 1 0 0 1 1
1 0 0 1
0 0 1 0 1 0
1 0 0 1
0 0 1 1 0 1
1 0 0 1
0 1 0 0 1
1 0 0 1
0 0 0 0 = R

Le reste R étant égal à 0, aucune erreur n'est détectée.

Annexe 4 AES (Advanced Encryption Standard)[

L'explication et les exemples concernant l'algorithme AES sont repris de [Swenson, 2008].
L'algorithme AES est un algorithme de chiffrement symétrique par blocs qui a été sélectionné par le gouvernement américain dans le but de remplacer le DES.
Il résulte de certaines restrictions apportées à l'algorithme Rijndael, ce dernier permettant d'utiliser des blocs et des clés de différentes tailles, AES n'autorise que des blocs de 128 bits avec des clés de 129, 192 et 256 bits [Swenson, 2008]. Dans un premier temps chaque bloc est réparti dans une matrice de quatre lignes et d'un nombre de colonnes compris en 4 et 8 de façon à ce que chaque cellule contienne 8 bits. Cette matrice est appelée l' « état ».

Cet algorithme est constitué de 4 opérations principales :

1. SubBytes : l'état est - comme mentionné ci-dessus - constitué de cellules de 8 bits. Cette opération consiste à substituer les 8 bits de chacune des cellules de l'état par 8 autres bits spécifiés par une fonction faisant correspondre une valeur de sortie sur 8 bits aux 2^8 entrées possibles. Cette fonction est la « S-Box » de Rijndael et peut se définir comme une matrice dont la construction dépasse la portée de ce travail.

2. ShiftRows : durant cette opération, la seconde ligne de l'état du bloc subit un décalage cyclique de 8 bits vers la gauche. La troisième subit un décalage identique mais de 16 bits et la troisième de 24 bits.

3. MixColumns : cette méthode fait appel à la notion de corps fini que l'on définit : « Dans le cadre de l'arithmétique modulo n, si n est premier ou une puissance première d'un nombre premier, alors nous avons un **corps fini** »[Buchmann, 2006]. Cette méthode fait usage de la multiplication sur les corps finis notées « x ». « Dans le cadre de AES cela consiste à multiplier deux nombre de 8 bits entre eux et à calculer le reste de la « division binaire longue » de ce produit par 283. Cette multiplication sera utilisée afin de multiplier chacune des colonnes de l'état par la matrice suivante
4.

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

Tableau 1: Matrice chiffrement

Après la multiplication matricielle des quatre colonnes, le nouvel état est obtenu.

5. AddRoundKey : durant cette opération, chaque cellule de l'état subit un XOR avec les 8 bits correspondants dans la matrice obtenue à partir de la clé.

Ces méthodes (excepté AddRoundKey) sont associées à une méthode inverse utilisée pour le déchiffrement.

1. InvSubBytes : utilisations de la fonction inverse.

2. InvShiftRows : le décalage se fait vers la droite.

3. MixColumns : la matrice utilisée est la suivante :

15	11	14	9
9	15	11	14
14	9	15	11
11	14	9	15

Les opérations de chiffrement et déchiffrement correspondent à une séquence de ces méthodes définie dans le tableau ci-dessous :

	Chiffrement	Déchiffrement
1	AddRoundKey	AddRoundKey
2	Répéter un nombre de fois dépendant de la taille de la clé et des blocs : 1. SubBytes 2. ShiftRows 3. MixColumns 4. AddRoundKey	Répéter un nombre de fois dépendant de la taille de la clé et des blocs : 1. InvSubBytes 2. InvShiftRows 3. InvMixColumns 4. InvAddRoundKey
3	SubByte	InvSubyte
4	ShiftRows	InvShiftRows
5	AddRondKey	AddRoundKey

Tableau 2: Chiffrement/Déchiffrement AES

Finalement AES propose deux méthodes utilisées afin d'étendre la clé :

1. SubWord : cette méthode prend 32 bits en entré, les divise en 4 blocs de 8 bits auxquels elle applique la méthode SubBytes. Le résultat est la concaténation du résultat des 4 appels.

2. RotWord : cette méthode prend 32 bits en entrée, les divise en 4 blocs de 8 bits et décale de façon cyclique vers la gauche chaque bloc ainsi obtenu.